

**Vysoká škola ekonomická v Praze**

**Fakulta informatiky a statistiky**

**Katedra informačních technologií**

Studijní program: Aplikovaná informatika

Obor: Informační systémy a technologie

# **Zákaznický vývoj v systému SAP**

## **DIPLOMOVÁ PRÁCE**

Student : Josef Rybička

Vedoucí : doc. Ing. Alena Buchalceková, Ph.D.

Oponent : doc. Ing. Jan Pour, CSc.

**2013**

## **Prohlášení**

Prohlašuji, že jsem diplomovou práci zpracoval samostatně a že jsem uvedl všechny použité prameny a literaturu, ze které jsem čerpal.

V Praze dne 6. května 2013

.....  
Josef Rybička

## **Poděkování**

Rád bych využil této příležitosti a poděkoval vedoucí práce, doc. Ing. Aleně Buchalceové, Ph.D., která mi pomohla vybrat téma, poskytla užitečné rady při jeho zpracování a zajistila konzultaci ve firmě SAP ČR. V souvislosti s tím také děkuji panu Gabrielu Tunovi za zadání ukázkové aplikace v jazyce ABAP, za cenné rady při jejím vývoji a za čas, který mi věnoval během konzultací. Na závěr děkuji všem, kteří mě podporovali po celou dobu studia, zejména své rodině.

## **Abstrakt**

Tato práce se zabývá systémem SAP z technologického pohledu a zaměřuje se na technologický základ SAP NetWeaver. Hlavním cílem práce popsat a prakticky demonstrovat zákaznický vývoj v systému SAP a v souvislosti s tím i charakterizovat technologický základ SAP NetWeaver a jeho strategický význam pro podniky. Vývoj je demonstrován vytvořením aplikace v jazyce ABAP s použitím metodiky MMSP. Práce je uceleným materiálem pro začínající vývojáře v systému SAP a obsahuje návody, jak aplikace vyvíjet a jak používat vývojové nástroje.

## **Klíčová slova**

SAP, ABAP, SAP NetWeaver, SAP NetWeaver Application Server, MMSP, Zákaznický vývoj

## **Abstract**

This thesis deals with SAP system from technological point of view and focuses on SAP NetWeaver platform. The main goal of this work is to describe and demonstrate custom development in SAP system. Secondary goal is to provide an overview of SAP NetWeaver platform and its capabilities and its important value for companies. The development is shown by creating an application in ABAP language and by using MMSP method. This thesis is complete guide for beginning SAP developers and contains tutorials about programming in ABAP and using its development tools.

## **Keywords**

SAP, ABAP, SAP NetWeaver, SAP NetWeaver Application Server, MMSP, Custom development

# Obsah

<b>1</b>	<b>Úvod</b>	<b>1</b>
1.1	Motivace .....	2
1.2	Důvod výběru tématu práce .....	2
1.3	Cíle a přínosy .....	3
1.4	Struktura .....	3
1.5	Východiska práce .....	4
1.6	Rešerše české a zahraniční literatury .....	5
<b>2</b>	<b>Úvod do systému SAP</b> .....	<b>7</b>
2.1	ERP Systémy .....	7
2.1.1	Nové formy podnikových aplikací .....	8
2.1.2	Význam ERP systémů pro podniky .....	8
2.1.3	Trh s ERP systémy v ČR .....	9
2.2	Společnost SAP .....	10
2.2.1	Historie společnosti SAP .....	10
2.2.2	Postavení na trhu společnosti SAP .....	11
2.2.3	Produkty společnosti SAP .....	13
2.3	Architektura systému SAP .....	14
2.3.1	Komponenty .....	14
2.3.2	Moduly a podnikové scénáře .....	15
2.3.3	Transakce .....	15
2.3.4	Klienti .....	16
2.4	Technologická architektura .....	16
2.5	Odvětvová řešení .....	17
2.6	SAP a architektura orientovaná na služby .....	18
2.6.1	SOA .....	18
2.6.2	Adaptace společnosti SAP na SOA .....	19
<b>3</b>	<b>SAP NetWeaver</b> .....	<b>21</b>
3.1	Důvody vzniku platformy SAP NetWeaver .....	21
3.2	Od webového aplikačního serveru k SAP NetWeaver .....	24
3.3	Schéma platformy SAP NetWeaver .....	26
3.4	Oblasti SAP NetWeaver .....	27
3.4.1	Řízení základu (Foundation Management) .....	28
3.4.2	Middleware .....	29
3.4.3	Řízení informací (Information Management) .....	30
3.4.4	Produktivita týmu (Team Productivity) .....	31
3.4.5	Kompozice (Composition) .....	32
3.4.6	Řízení podnikových procesů (Business Process Management) .....	33

3.5	Případy užití platformy SAP NetWeaver.....	34
3.6	Stavební bloky platformy SAP NetWeaver.....	34
3.6.1	Druhy užití (Usage Types).....	35
3.6.2	Samostatné stroje (Standalone Engines).....	35
3.6.3	Klienti (Clients).....	36
3.6.4	Mapování stavebních bloků na případy užití .....	36
3.7	Shrnutí přínosů platformy SAP NetWeaver .....	37
4	<b>Zákaznický vývoj v jazyce ABAP .....</b>	<b>40</b>
4.1	Zákaznický vývoj.....	40
4.1.1	ABAP .....	41
4.1.2	Java .....	41
4.1.3	Composition Environment .....	42
4.2	Proces zákaznického vývoje.....	42
4.3	Úvod do vývoje v jazyce ABAP.....	42
4.3.1	Vývoj na straně serveru.....	42
4.3.2	Třívrstvá architektura .....	43
4.3.3	Pracovní procesy .....	44
4.4	Práce v SAP NW AS ABAP.....	45
4.4.1	Správa objektů .....	46
4.5	Transportní systém.....	47
4.5.1	Typy klientů a systémů SAP .....	47
4.5.2	Princip přenosu objektů mezi systémy .....	49
4.6	Vývoj aplikačních programů .....	50
4.6.1	Struktura programů v jazyce ABAP.....	51
4.6.2	Deklarace proměnných.....	53
4.6.3	Dialogové programování .....	53
4.6.4	Programování reportů .....	54
4.7	Datový slovník.....	55
4.7.1	Datové prvky .....	55
4.7.2	Datové modelování .....	56
5	<b>Ukázková aplikace .....</b>	<b>58</b>
5.1	Předpoklady a omezení ukázkové aplikace .....	58
5.2	Zadání ukázkové aplikace .....	59
5.3	Použití metodiky .....	59
5.4	Fáze zahájení.....	60
5.4.1	Požadavky .....	60
5.4.2	Případy užití.....	61
5.5	Fáze rozpracování.....	62
5.6	Fáze konstrukce.....	64
5.7	Fáze zavedení.....	67

## **6 Závěr 68**

**Terminologický slovník..... 70**

**Seznam literatury..... 73**

**Seznam obrázků a tabulek..... 76**

Seznam obrázků .....76

Seznam tabulek .....77

**Rejstřík 78**



# 1 Úvod

Trh celopodnikových transakčních aplikací (ERP systémů) poskytuje dnes tak širokou škálu produktů, že mnoho podniků může pro podporu svého byznysu implementovat ERP systémy v podobě typového aplikačního software (TASW, aplikační balík) a nemusí aplikace vyvíjet od začátku. Typový aplikační software je vyvinut na principu generalizace – tj. požadavky různých typů podniků jsou zobecněny tak, aby pokrývaly požadavky jednotlivých podniků (Voříšek 1997, s. 47). Přizpůsobení specifickým požadavkům podniku je pak realizováno nastavením parametrů tohoto software v procesu, který se označuje jako *customizace*<sup>1</sup>.

Ačkoliv je ERP systém v podobě TASW vysoce parametrický, je téměř nemožné, aby jeho dodavatelé zahrnuli všechny požadavky do jediného produktu. Proto mají dodavatelé ERP systémů různá řešení již předem upravená pro specifické odvětví (výroba, služby, finanční sektor, veřejný sektor). Přesto každý podnik (i v rámci odvětví) je jedinečný a má své vlastní požadavky na funkcionalitu. I po nasazení<sup>2</sup> TASW je často potřeba funkcionalitu ERP systému upravit nebo vytvořit zcela novou aplikaci, která je jeho součástí. Softwarová architektura ERP systémů pro tyto účely zahrnuje vlastní vývojové prostředí, tj. integrované vlastní programovací prostředky nebo jazyky (Gála et al. 2009, s. 161).

V ERP systému (a dalších podnikových aplikacích) od společnosti SAP jsou tyto prostředky součástí technologického základu, který se nazývá SAP NetWeaver.

Tvorba aplikací pomocí těchto prostředků za účelem pokrytí speciálních požadavků zákazníků se označuje jako **zákaznický vývoj** (*Custom development*) a je hlavním tématem této práce. Zákaznický vývoj zde též označuji jako **vývoj v systému SAP** – viz např.

V této práci se obecně zaměřuji na systém SAP z technologického hlediska a jeho možnostmi, jak reagovat na rychle se měnící požadavky na funkcionalitu v situaci, kdy ani konfigurace<sup>3</sup> nedokáže tuto flexibilitu systému zajistit. To znamená, že popisují technologický základ **SAP NetWeaver** a jeho **strategické využití**. Z oblasti zákaznického vývoje je práce zaměřena na komponentu **SAP NetWeaver Application Server**, konkrétně na jazyk **ABAP**.

---

<sup>1</sup> Pojem *customizace* z anglického *customization* samozřejmě není v češtině ustálen, nicméně zde a na dalších místech práce používám pojem dle (Voříšek 2011).

<sup>2</sup> Nasazení TASW dle (Voříšek 2011, s. 59) se skládá z *lokalizace* (funkcionalita se upravuje dle platné legislativy, národního jazyka a kulturních zvyklostí), *customizace* (přizpůsobení parametrů dle podnikových procesů) a *integrace* (propojení s ostatními systémy).

<sup>3</sup> Takto se označuje *customizace* v terminologii implementace systému SAP

## 1.1 Motivace

Změnit funkcionalitu TASW jinak než úpravou parametrů znamenalo dříve zasahovat do programového kódu aplikace, což bylo velmi nákladné a riskantní, protože aplikace k těmto změnám nebyly uzpůsobeny a s příchodem každé nové verze, bylo nutné změny programovat znovu (Voříšek 1997, s. 47). Proto jsou dnes aplikace systému SAP postaveny na takovém technologickém základu, který umožňuje **pružně reagovat** na měnící se hospodářské prostředí a na požadavky na změnu funkcionality od vedení podniku. Při zákaznickém vývoji se přizpůsobují standardní aplikace systému SAP potřebám zákazníka, vytváří se nové specializované aplikace nebo se např. tvoří rozhraní pro propojení se systémy od jiných dodavatelů.

Zákaznický vývoj má tedy velký význam pro úplné pokrytí **specifických požadavků podniku**. Podniky (zákazníci) mohou tento vývoj řešit pomocí interních zaměstnanců (*in-house development*), nebo pomocí certifikovaných SAP partnerů (SAP 2013b). První varianta je pro podnik jistě levnější, avšak problém je, že odborníků na programové úpravy systému SAP je velmi málo. To se také odráží na **platovém ohodnocení** těchto vývojářů, které je na mnohem vyšší úrovni než u vývojářů např. v jazyce Java (Tuna 2013).

Dalším důvodem, proč se tímto tématem zabývat, jsou **rostoucí požadavky na IT profesionály**. Na implementaci ERP systémů se podílí mnoho pracovníků z různých profesí – projektoví manažeři, konzultanti, specialisté na funkční oblasti podniku, vývojáři, testéři atd. Jak uvádí (Anderson 2012, s. 369), každá z těchto profesí se zaměřuje buď na podnikovou (byznys) nebo na technickou oblast při implementaci systému. Čím dál více žádaná je však taková skupina pracovníků, která má jak podnikové, tak technické znalosti. Proto je vhodné, aby i podnikově zaměření pracovníci měli alespoň základní přehled o technologických aspektech systému SAP.

Jako reakce na tento trend začínají ve světě vznikat **vysokoškolské studijní obory**, kde se studenti učí o informačních systémech z technického i podnikového pohledu. Jako příklad lze uvést *East Tennessee State University – Department of Computer and Information Sciences*. Zde byly postupně založeny skupiny kurzů, jako jsou Základy podnikových informačních systémů, Implementace informačních systémů a Programování v podnikových IS, která obsahuje kurzy vývoje v jazyce ABAP. Tato poslední skupina je studentům přístupná až po absolvování předchozích skupin, zaměřených na podnikové procesy (Pittarese 2012). Obdobně orientované kurzy mají jistě **velký význam pro studenty**, kteří uvažují o technickém uplatnění při implementaci informačních systémů a nestačí jim pouze teoretický pohled.

## 1.2 Důvod výběru tématu práce

Problematika vývoje v systému SAP mě velmi zajímá. Toto téma jsem si vybral zejména z toho důvodu, abych rozšířil oblasti svého uplatnění a také uspořádal informace nabyté

studiem systému SAP. Je to také směr, kterým bych se chtěl ubírat ve smyslu budoucího zaměstnání. Profese konzultanta či vývojáře při implementaci systému SAP je velkou výzvou, ale jak už jsem uvedl v motivačním odstavci, tato profese je vzhledem k nedostatku odborníků velmi žádaná. Téma jsem si vybral také z důvodu jeho přínosů – viz dále.

## 1.3 Cíle a přínosy

Cílem této práce je popsat a prakticky demonstrovat zákaznický vývoj v systému SAP. Tohoto cíle by mělo být dosaženo splněním následujících dílčích cílů:

- popsat základní architekturu a principy systému SAP,
- charakterizovat technologický základ SAP NetWeaver a jeho strategické využití
- popsat vývoj v jazyce ABAP a nezbytné nástroje, které tento vývoj umožňují
- vyvinout ukázkovou aplikaci v jazyce ABAP

Práce by měla být uceleným materiálem, který poskytuje úvod do vývoje aplikací v systému SAP, včetně praktické ukázky programování v jazyce ABAP. Takto ucelený úvod, včetně praktické ukázky, se v tištěných a elektronických zdrojích příliš nevyskytuje (viz kap. 1.6) a už vůbec ne v českém jazyce. Práce tak ušetří mnoho času vývojářům, kteří s programovými úpravami systému SAP začínají.

Práce je určena jak pro začínající vývojáře v jazyce ABAP, tak i pro další členy týmu pracující na implementaci systému SAP, kteří se technickou stránkou primárně nezabývají, chtějí porozumět zdrojovému kódu jazyka ABAP a být schopni sami něco naprogramovat. Proto dílčí cíle zahrnují jak podnikové, tak technologické aspekty systému SAP. Práce by také mohla být podkladovým materiálem pro případnou výuku podobně zaměřeného kurzu na VŠE a obohatit tak poměrně omezenou zásobu literatury v češtině.

## 1.4 Struktura

**Teoretickou část** práce tvoří kapitoly 2 a 3, ve kterých se zabývám systémem SAP od obecných charakteristik až po softwarovou architekturu a technologický základ SAP NetWeaver. V této části by měl čtenář získat přehled o tom, jaký je význam podnikových informačních systémů, které aplikace tvoří systém SAP, jaké jsou jeho současné trendy a jaký je význam technologického základu SAP NetWeaver.

**V praktické části** popisují principy zákaznického vývoje v jazyce ABAP, což je tématem kapitoly 4. Charakterizují zde jednak samotný jazyk a také nástroje, ve kterých vývoj probíhá. V 5. kapitole pak popisují, jak jsem vyvíjel ukázkovou aplikaci pomocí metodiky MMSP.

## 1.5 Východiska práce

Jedním ze zdrojů znalostí potřebných pro vznik této práce byly odborné knihy zabývající se systémem SAP. Pomohly mi zejména při zpracování teoretické části. Jedná se o zdroje (Anderson 2012; Kühnhauser 2012; Woods, Word 2004a) viz rešerše literatury. Kromě toho jsem měl možnost konzultovat programování aplikace s vývojářem společnosti SAP ČR, který mi také poskytl několik výukových materiálů a navrhl zadání aplikace. Mimo to jsem čerpal z oficiální dokumentace pro SAP NetWeaver<sup>4</sup> a zejména z diskuzních skupin sociální sítě, která sdružuje profesionály v oblasti systému SAP – SCN (*SAP Community Network*)<sup>5</sup>.

Tato práce se týká **informačních systémů** (IS). Tento pojem lze definovat jako „*systém pro sběr, přenos, uchování, zpracování a poskytování dat (informací, znalostí) využívaných při činnosti podniku. Jeho komponentami jsou informační a komunikační technologie, data a lidé*“ (Voříšek 2011). V této definici je akcentováno, že se jedná o podnikový informační systém, což je důležité, protože IS se nemusí nutně týkat pouze podniku. Informační a komunikační technologie (ICT) se skládají z hardwarové a softwarové složky, které obsahují další komponenty. Vzhledem k zaměření této práce je nejdůležitější softwarovou komponentou aplikační softwaru, který je určen pro podporu podnikových procesů, individuální a skupinovou práci uživatelů a poskytování informací zpracovávaných v rámci různých oddělení v podniku. Jádrem podnikového aplikačního softwaru bývá ERP (*Enterprise Resource Planning*), který pomáhá plánovat podnikové zdroje a podporuje hlavní podnikové procesy (Voříšek 2011, s. 20). Z hlediska způsobu vzniku se aplikační software dělí na individuální aplikační software (IASW), který je vyvíjen „na míru“ podle požadavků podniku, a na typový aplikační software (TASW), který obecně pokrývá všechny požadavky určitého typu podniků a lze přizpůsobit pomocí nastavení parametrů.

Vymezit je také třeba samotný pojem „**SAP**“, který se často používá v různých významech. Zkratka SAP znamená *Systémy Aplikace a Produkty* a jedná se o německou společnost sídlící ve Walldorfu (SAP 2013e). Originální název zní *Systems, Applications and Products in Data Processing* (Anderson 2012).

Nicméně pojmem SAP se také často označuje obecně jakýkoliv produkt, který společnost SAP dodala do podniku, aniž by se uvedla konkrétní skupina aplikací. V teoretické části pojmem „systém SAP“ označuji celopodnikové řešení, které se nazývá *SAP Business Suite* (viz kap. 2.2.3), ovšem bude-li to nutné, tak použiji název konkrétního produktu společnosti SAP.

---

<sup>4</sup> <http://help.sap.com>

<sup>5</sup> <http://scn.sap.com>

Nejrozsáhlejší komponentou systému SAP je produkt SAP ERP a je typickým příkladem **ERP systému**. Společně s dalšími komponentami se řadí mezi **typový aplikační software** (TASW)

**SAP NetWeaver** je současný technologický základ produktů od společnosti SAP a umožňuje jejich provoz a rozvoj. Na tomto základě jsou postaveny jednotlivé komponenty podnikových řešení. Obsahuje nástroje pro rozšíření systému SAP např. do prostředí internetu a k propojení s ostatními aplikacemi mimo systém SAP. Podporuje vývoj aplikací v jazyce ABAP nebo Java.

## 1.6 Rešerše české a zahraniční literatury

Z oblasti systémů SAP lze nalézt převážně postupy a případové studie projektů implementace celého systému nebo jeho částí, uživatelské příručky nebo literaturu zaměřenou na koncové uživatele. Mnohem méně zdrojů je však z oblasti technologického základu SAP NetWeaver a programovacího jazyka ABAP, tedy zdrojů technického zaměření.

Provedl jsem analýzu **zahraniční literatury** a kromě dokumentace všech technologií, které systém SAP používá<sup>6</sup>, je nejznámějším zdrojem kniha *SAP NetWeaver For Dummies* (Woods, Word 2004a). Jedná se o shrnutí všech možností, které SAP NetWeaver nabízí. Jak už název napovídá, kniha je napsána jednoduchou a zábavnou formou, aby byl této platformě schopen kdokoli porozumět. Kniha však neobsahuje praktické příklady užití, tedy jak danou problematiku naimplementovat. Méně známou alternativou této knihy je verze pro odborníky (Woods, Word 2004b). Dalším zdrojem je kniha *ABAP – výukový kurz* (Kühnhauser 2012), která seznamuje čtenáře s jazykem ABAP. Z této publikace budu vycházet v části, kde představím tento programovací jazyk a jeho užití. Bude mi sloužit také jako příručka při tvorbě aplikace. Kniha je ale pouze referenční příručkou a neposkytuje příliš dobrý návod, jak zákaznický vývoj probíhá. Posledním významným zdrojem, je průřezová kniha *Naučte se SAP za 24 hodin*<sup>7</sup> (Anderson 2012). Tato kniha se týká vývoje jen okrajově a je určena převážně pro manažery. Od stejného autora a jeho kolektivu projektových manažerů existuje také kompletní průvodce implementací celého systému SAP s názvem „*SAP Implementation Unleashed*“ (Anderson 2009). Zde se kolektiv autorů zaměřuje na celý proces implementace (např. partnerskou firmou). Z oblasti zákaznického vývoje je zde pouze popsána instalace aplikačního serveru pro vývoj v jazyce ABAP.

---

<sup>6</sup> dostupných na serveru SAP Help nebo v podobě knih z nakladatelství SAP Press

<sup>7</sup> Jedná se o marketingový název. Ve skutečnosti kniha obsahuje 24 lekcí, kde by každá lekce mohla být s určitým úsilím pochopena za hodinu.

Míru pokrytí mého tématu **v české literatuře** jsem analyzoval na serveru Vysokoškolské kvalifikační práce<sup>8</sup> – zaměřil jsem se tedy na bakalářské a diplomové práce. Nalezl jsem celkem 5 prací, které se zabývají mým tématem, ovšem z různých pohledů.

Autoři Roman Pavlovský a Jan Tyle, zpracovali společně téma *Aplikace na správu projektů pro SAP NetWeaver Portal*. Zaměřují se na vývoj portálové aplikace, a to s využitím technologie Java pro aplikační logiku (Pavlovský 2011) a technologie Adobe Flex pro uživatelské rozhraní (Tyle 2011). Moje práce obohacuje tuto oblast zaměřením na druhý programovací jazyk platformy NetWeaver – na jazyk ABAP.

Autor Slavomil Saska zpracoval téma *Implementace ukázkové aplikace v kontextu SAP NetWeaver platformy* (Saska 2010). V praktické části představil krátké případové studie vývoje aplikace s využitím technologií Visual Composer a Java Web Dynpro. Autor se zabývá vývojem aplikace s webovým rozhraním. Moje práce využívá rozhraní SAP GUI.

Vývojem v podnikových aplikacích se také zabývá Jan Chaloupka ve své práci *Vývoj specializovaných aplikací pro MS Dynamics AX* (Chaloupka 2009) popisuje prostředí MorphX. Zde jsem se inspiroval strukturou práce – seznámení s vývojovým prostředím a vývojem konkrétní aplikace.

Poslední práce se nejvíce týká mé teoretické části, a sice *Technologie v rámci mySAP ERP* (Koubek 2006). Autor analyzuje technologii a architekturu vývojové prostředí tehdejšího systému SAP R/3 a popisuje technologie pro tvorbu výkazů v tomto systému. Jde však o popis a nikoliv užití těchto technologií. Kromě toho byla práce napsána před sedmi lety, a tím pádem je dnes již mírně zastaralá. Přínosem mojí práce je zde zaktualizování teorie a doplnění o praktické užití jazyka ABAP.

---

<sup>8</sup> <http://www.theses.cz>

## 2 Úvod do systému SAP

V této kapitole charakterizují obecně ERP systémy, a jejich strategický význam. Dále popisují společnost SAP a zařazení systému SAP ERP do jejího portfolia podnikových řešení.

Cílem je seznámit čtenáře s tematikou práce z širšího hlediska a vytvořit tak znalostní základ pro lepší pochopení všech následujících kapitol.

### 2.1 ERP Systémy

Nejrozsáhlejším produktem firmy SAP je systém SAP ERP, který patří mezi celopodnikové transakční aplikace – tzv. ERP systémy.

ERP (*Enterprise Resource Planning*) lze definovat jako „softwarová řešení užívaná k řízení podnikových dat a pomáhající plánovat celý logistický řetězec od nákupu přes sklady po výdej materiálu, řízení obchodních zakázek od jejich přijetí až po expedici včetně plánování vlastní výroby a s tím spojené finanční a nákladové účetnictví i řízení lidských zdrojů“ (Basl, 2008). ERP systém má tedy logistickou část (nákup, výroba, distribuce) a ekonomickou část (finanční a nákladové účetnictví).

ERP systém je hlavním představitelem **celopodnikových transakčních systémů**, které pokrývají většinu podnikových procesů a funkcí. Transakční se nazývají proto, že realizují obchodní, finanční a další transakce – vkládání zákazníků, zboží, dodavatelů – nad datovou základnou. Tyto systémy a technologie, které využívají, jsou optimalizovány pro vkládání dat s minimální dobou odezvy.

Hlavním účelem ERP systémů je tedy **zpracovávání transakcí a automatizace podnikových procesů**. Právě automatizace je jeden z mnoha pohledů při definování ERP systému, které uvádí (Basl, 2008). Dalším pohledem je např. **datový pohled**, který zdůrazňuje, že ERP systém je tvořen jedinou datovou základnou, která zabraňuje tvorbě duplicitních dat. V době izolovaných podnikových aplikací se pracovalo s daty separátně např. v rámci jednotlivých oddělení pomocí databází, které nebyly vzájemně integrované. Tím rostla nekonzistence, chybovost a náklady na údržbu těchto nepropojených aplikací. Pro ERP systémy je charakteristické, že představují **jednu konzistentní aplikaci**, která poskytuje informatickou podporu všem důležitým procesům ve firmě.

Samotná zkratka ERP doslova znamená „plánování podnikových zdrojů“. V současné době však funkcionality ERP mnohokrát převyšuje funkcionalitu odpovídající tomuto názvu (Gála et al. 2009, s. 159). Dochází tak ke slučování dalších aplikací a nástrojů např. CRM, SCM, portálové řešení, analytické aplikace apod. Tím vznikají komplexní balíky založené na jádru ERP a tvoří tzv. ERP II.

### 2.1.1 Nové formy podnikových aplikací

S vývojem technologií se také postupně rozšiřovaly i formy podnikových aplikací. Od poloviny 20. století existovaly pouze izolované aplikace instalované na sálových počítačích, další etapou byla éra aplikací s architekturou klient/server a od 21. Století začaly vznikat systémy formou „on-demand“ neboli *Software As A Service (SaaS)*.

Poskytování ERP systémů jako služby funguje na principu předplatného. Oproti klasickým systémům, kde se platí za licenci na jednotlivé komponenty, zde firmy platí pouze za to, co skutečně využijí. V tomto obchodním modelu poskytovatele dochází k úsporám z rozsahu, protože stejnou aplikaci používá mnoho uživatelů. Z toho důvodu je tato forma ERP systému cenově dostupnější a snižuje náklady na vlastnictví systému (TCO) protože podnik se vůbec nemusí starat o jeho provoz. O veškerou infrastrukturu, údržbu, zálohování a aktualizace se stará poskytovatel. Podnik pouze čerpá funkcionalitu služby.

SaaS je považován za další etapu vývoje podnikových systémů a je možné, že v budoucnu nahradí formu klient/server, která v současnosti převažuje.

### 2.1.2 Význam ERP systémů pro podniky

Nasazením ERP systému může podnik získat podstatnou konkurenční výhodu, a to např. díky vnitropodnikové integraci a integraci podniku s dodavateli nebo zákazníky, což rozšířená řešení ERP II umožňují. Díky integraci dochází k zefektivnění procesů, a ke snižování nákladů v podniku. Právě snižování nákladů je jedna z nejdůležitějších věcí, na které manažeri podniku „slyší“ když jsou přesvědčováni o zavedení nového systému. Mohou totiž díky informačním technologiím získat konkurenční výhodu v podobě prvenství v nákladech. ERP systém dokáže snížit náklady zejména díky poskytování včasných a správných informací, jak uvádí (Basl 2008, s. 177):

- Přesná znalost termínů objednávek a dodávek a jejich stavu může redukovat pojistný stav zásob.
- Díky vzájemnému informování mezi podnikem, dodavateli a zákazníky lze předcházet zbytečným časovým rezervám a dodržovat termíny dodávek.
- Elektronické transakce (objednávky, bankovníctví, zprávy) zrychlují komunikaci vně i uvnitř podniku.
- Informace z prodejů (díky aplikaci CRM) pomáhají poznat potřeby zákazníků, aniž by bylo nutné provádět nákladná marketingová šetření.

Předcházející přínosy mohou platit pouze za předpokladu úspěšné implementace systému a snahy všech zúčastněných uživatelů sdílet informace a efektivně využívat funkce systému.



Efektů plynoucích ze zavedení ERP systémů do podniku lze nalézt více. Uspořádání dle BSC (*Balanced Scorecard*) je uvedeno např. v (Basl 2008, s. 180), nebo též v (Gála et al. 2009, s. 184)

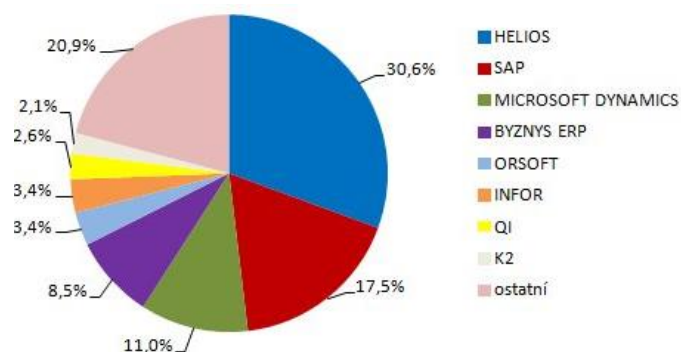
### 2.1.3 Trh s ERP systémy v ČR

Penetrace **českého trhu** ERP systémy je stále nízká. Dokládá to studie provedená v roce 2012 Českým statistickým úřadem. Oproti roku 2009 je však situace lepší, jak ukazuje následující tabulka.

**Tabulka 1:** *Nasycenost trhu s ERP produkty v ČR na straně poptávky. Zdroj dat: (ČSÚ 2012; Sodomka, Klčová 2012)*

rok	2009	2011	2012
Podniky celkem	13,9 %	25,2 %	24 %
Malé podniky	8,2 %	17,8 %	16 %
Střední podniky	30,2 %	50,3 %	51 %
Velké podniky	59,7 %	79,6 %	79 %

Důvodem nízké penetrace může být roztržitost na straně nabídky, která je způsobena tím, že se na našem trhu **neuskutečňuje** příliš mnoho **akvizic**, jako v zahraničí. Trh ERP systémů pro malé podniky (od 10 do 49 zaměstnanců) ovládají pouze **tuzemští výrobci** – leaderem je společnost Asseco Solutions se svým systémem *Helios Orange*. Na segment středních podniků (od 50 do 249 zaměstnanců) již začínají pronikat i zahraniční výrobci, avšak stále nejsou na vedoucích pozicích. Ti se začínají dostávat do popředí až v segmentu **velkých podniků**. I zde ale leaderem zůstává systém Helios. Obrázek 1 ukazuje zastoupení jednotlivých výrobců ERP systémů na českém trhu v segmentu velkých podniků. Hodnoceno bylo celkem 62 ERP systémů typu *All-In-One* zavedených do roku 2011 (Sodomka, Klčová 2012).

**Obrázek 1:**

*Zastoupení jednotlivých výrobců ERP systémů na českém trhu v segmentu velkých podniků. Zdroj: (Sodomka, Klčová 2012)*

## 2.2 Společnost SAP

Seznámení se systémem SAP zahájím stručnou charakteristikou společnosti, která tento systém vyvíjí již přes 40 let.

Společnost SAP je největším poskytovatelem podnikových aplikací a jednou z největších softwarových firem na světě. Víze společnosti je změnit svět, a to poskytováním integrovaných systémů, které pokrývají všechny potřeby podniků. Zaměřuje se na všechny softwarové produkty, které nějakým způsobem podporují podnikové procesy – software pro řízení podniku, nástroje Business Intelligence, systémy řízení vztahu se zákazníky, dále technologie, jako jsou platformy pro vývoj standardních i webových aplikací, řešení pro cloud computing apod. Portfolio produktů je tedy velmi široké a nemá smysl na tomto místě uvádět jeho úplný výčet. Pro zjednodušení se zaměřím pouze na kategorii celopodnikových řešení – viz dále.

### 2.2.1 Historie společnosti SAP

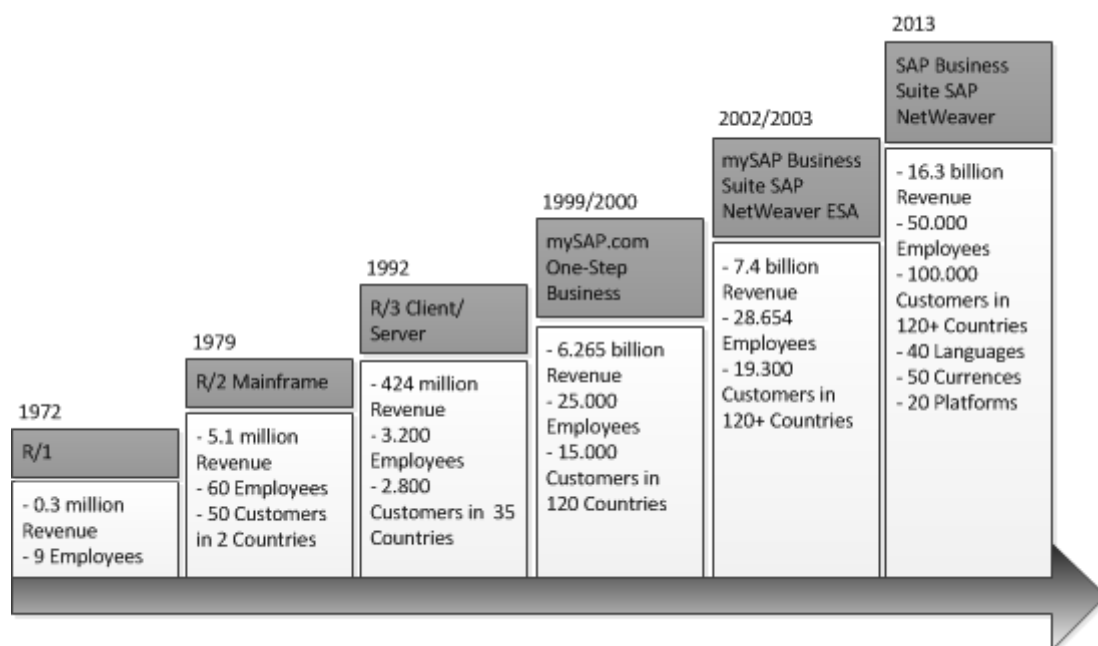
Společnost SAP byla založena v roce 1971 pěti bývalými zaměstnanci IBM, kteří získali prvotní verzi aplikace od firmy Big Blue výměnou za 8% zakladatelských akcií IBM (Chickowski 2011). V průběhu 70. let se společnost SAP zabývala vývojem systému SAP R/2, určeným pro mainframe počítače. Později se však odklonila od tohoto technologického modelu a začala navrhovat systémy tak, aby je bylo možné provozovat na kterékoliv platformě, což bylo považováno za zlom ve vývoji podnikových aplikací. Tím se dostala společnost SAP mezi hlavní soupeře na trhu informačních technologií a své postavení si ukovala od 90. let se svým produktem SAP R/3 postaveným na architektuře klient/server.

SAP R/3 byl prvním produktem společnosti SAP, který bylo možné označit jako OLTP (*OnLine Transaction Processing*) systém, tedy systém pro plnění každodenních transakčních požadavků (Anderson 2012). V té době skutečně bylo jedno a totéž, když se řeklo

SAP nebo SAP R/3. Oba termíny označovaly produkt firmy SAP, protože žádný jiný produkt nebyl. Následovníkem SAP R/3 byl systém SAP ERP.

Dříve bylo možné se setkat s názvem *mySAP ERP*. V roce 2007 však došlo ke změně názvu a odstranění předpon „my“. Phillip Say, viceprezident pro marketing a prodej v Evropě, to označil jako přežitek z éry „dot com“ firem, a proto se společnost rozhodla produkty přejmenovat (Farber 2007).

Obrázek 2 ukazuje, jak se vyvíjela společnost SAP a její produkty od jejího založení v roce 1972.



**Obrázek 2:**

*Evoluce podnikových aplikací společnosti SAP. Zdroj: upraveno dle (Woods, Word 2004a, s. 29)*

## 2.2.2 Postavení na trhu společnosti SAP

Společnost SAP má mnoho konkurentů, avšak v určitém smyslu se tyto konkurenční firmy podporují navzájem. SAP např. považuje společnost Oracle za dodavatele databází pro své produkty, společnost Microsoft za dodavatele operačních systémů, na kterých produkty od SAPu běží apod.

Nicméně dle výzkumu firmy Gartner je společnost SAP nyní lídrem na trhu v oblasti ERP řešení pro podniky s obratem od 10 milionů do 1 miliardy dolarů, tedy pro středně velké podniky (Hestermann et al. 2012). Mezi lídry si zároveň drží vedoucí pozici (viz magický kvadrant od firmy Gartner – Obrázek 3). Dále je lídrem na trhu v oblasti systémů Business

Intelligence a platformem pro provádění analýz (Schlegel et al., 2013) a také platformem pro vývoj mobilních aplikací (Clark et al. 2012).



**Obrázek 3:**

*Postavení na trhu společnosti SAP v oblasti ERP řešení pro středně velké podniky. Zdroj: (Hestermann et al. 2012)*

Pro představu o firmě SAP, jako jedné z největších softwarových firem na světě, uvádím tabulku obrátů za fiskální rok 2012. Následující tabulka obsahuje vývoj obrátů pro jednotlivé kvartály a v rámci nich jsou rozepsány obraty pro jednotlivé skupiny produktů. Všechna uvedená finanční data neodpovídají standardům IFRS<sup>9</sup>.

**Tabulka 2:** *Údaje o obratech společnosti SAP za jednotlivé kvartály roku 2012. Zdroj dat převážně z: (Brandt 2013)*

Obrat (mil. €)	1. kvartál	2. kvartál	3. kvartál	4. kvartál	Celkem
Software (licence)	637	1059	1026	1936	<b>4658</b>
Podpora	1954	2014	2106	2170	<b>8244</b>
Předplatné SaaS	35	69	80	159	<b>342</b>

<sup>9</sup> *International Financial Reporting Standards*

Obrat (mil. €)	1. kvartál	2. kvartál	3. kvartál	4. kvartál	Celkem
<b>Celkem software a podpora</b>	<b>2626</b>	<b>3142</b>	<b>3212</b>	<b>4265</b>	<b>13245</b>
Služby	774	731	758	765	<b>3058</b>
<b>Celkový obrat</b>	<b>3357</b>	<b>3916</b>	<b>3970</b>	<b>5030</b>	<b>16303</b>

Z tabulky vyplývá rostoucí zájem zákazníků o produkty dodávané formou SaaS, což odpovídá trendům, které uvádí např. (Voříšek 2011) nebo (Basl 2008)

Za rok 2012 měla společnost SAP **celkový obrat 16,3 miliard €**, což je o 14% více, než v roce 2011. V oblasti položky **software** zaznamenala společnost růst **13%** a převýšila tak svá vlastní očekávání. Celkový **provozní zisk** za rok 2012 činil 5,02 miliard € (Brandt 2013, s. 5). Pro srovnání uvádím obrat společnosti Microsoft v oblasti podnikových aplikací, který v roce 2012 činil po přepočtu **18,7 miliard €**. Toto číslo však zahrnuje i obrat z kancelářských balíků MS Office (Helland 2012).

V České republice však systém společnosti SAP vedoucí pozici nemá, a to ani v řešeních pro velké podniky. V ČR je lídrem systém značky *Helios* s podílem 30% trhu. Teprve na druhém místě je SAP s podílem 17,5%. Toto číslo může být ale ovlivněno tím, že šetření se zaměřovalo pouze na produkty typu *All-in-One*, tím pádem by statistika zahrnovala pouze velké podniky, které mají SAP All-In-One. Z toho důvodu by mohlo být pravdivé i tvrzení, že „*SAP pokrývá 61% trhu TOP 100 českých firem*“, jak uvádí (SAP 2013f)

### 2.2.3 Produkty společnosti SAP

Společnost SAP nabízí širokou škálu softwarových produktů (SAP 2013d):

- podnikové aplikace,
- databáze a technologie,
- software pro tvorbu analýz
- cloudové služby (aplikace SaaS, cloudové platformy, virtualizace)
- mobilní technologie (mobilní aplikace a platformy pro jejich vývoj)

V této práci se zaměřuji na podnikové informační systémy, a tak popíši pouze nabídku podnikových aplikací. Nejznámějším celopodnikovým řešením je **SAP Business Suite** (viz dále), určené pro velké podniky. Toto řešení lze považovat za rozšířené ERP (ERP II), protože mimo jádra (ERP systému) obsahuje i další komponenty. Bylo by však chybné se domnívat, že společnost SAP se zaměřuje pouze na velké podniky. Vedle SAP Business Suite existují i řešení pro malé a střední podniky a firma SAP je dokonce v této oblasti na prvním místě mezi lídry ve světě. Jedná se o tyto produkty:

- SAP Business One,
- SAP Business ByDesign (poskytované jako SaaS),
- SAP All-In-One

Následující tabulka shrnuje vlastnosti řešení pro malé a střední podniky. Všechna řešení vychází z hlavního produktu SAP Business Suite. Liší se však počtem komponent a modulů, způsobem nasazení apod.

**Tabulka 3:** Porovnání řešení společnosti SAP pro malé a střední podniky. Zdroj: (Anderson 2012, s. 80)

Řešení	SAP Business One	SAP Business ByDesign	SAP Business All-In-One
Stručný popis	Jediná, integrovaná aplikace k řízení celého podniku	To nejlepší ze SAPu, dodávané formou SaaS	Komplexní, integrované a snadno přizpůsobitelné
Počet uživatelů	Do 100	100-500	Do 2500
Dostupné pro	40 zemí	USA, Velká Británie, Německo, Francie, Čína	50 zemí
Způsob nasazení	Vlastní zařízení zákazníka	<i>On-demand</i>	Vlastní zařízení zákazníka či hostované zařízení
Doba nasazení	2-8 týdnů	4-8 týdnů	8-16 týdnů
Objem transakcí	Nízký	Střední	Vysoký
Odvětvová řešení	Několik	Málo	Mnoho

## 2.3 Architektura systému SAP

Nyní se již podrobněji zaměřím na řešení **SAP Business Suite**, jeho uspořádání jednotlivých částí a vztahů mezi nimi.

### 2.3.1 Komponenty

Systém SAP je založen na principu specializace a integrace. Celé řešení SAP Business Suite, určené pro velké podniky, se skládá z komponent (aplikací). Každá komponenta odpovídá určitým podnikovým potřebám. Jádrem je SAP Enterprise Resource Planning

(SAP ERP), která např. pomáhá řídit finance a zdroje. Mezi komponenty, které SAP ERP rozšiřují, patří

- SAP Customer Relationship Management – řízení vztahu se zákazníky,
- SAP Product Lifecycle Management – řízení životního cyklu produktů,
- SAP Supply Chain Management – řízení dodavatelského řetězce,
- SAP Supplier Relationship Management – řízení vztahů s dodavateli.

### 2.3.2 Moduly a podnikové scénáře

**Moduly** jsou dílčí části komponent a řeší specifické podnikové oblasti. Každý modul je zaměřen na tu problematiku, která je specifická např. pro oddělení v organizační struktuře podniku. Kombinace modulů je schopna pokrýt určité *podnikové scénáře*. **Podnikový scénář** je pojem, kterým společnost SAP označuje souhrn podnikových procesů pro danou oblast, která může být podpořena moduly. Pro systém ERP uvádí (SAP 2013a) následující řešení podnikových scénářů:

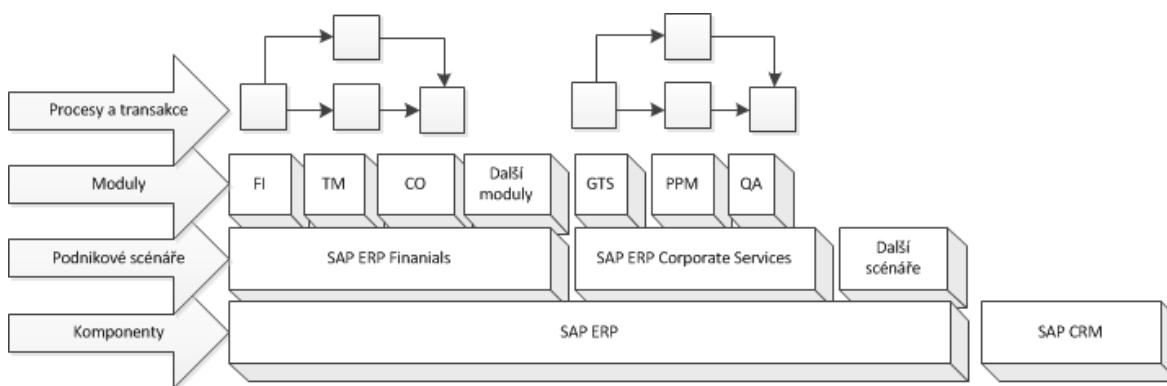
- **ERP Financials** – řízení podnikových financí. Obsahuje moduly Financial Accounting (FI), Treasury and Risk Management (TM), Controlling (CO) a další.
- **ERP Human Capital Management** – správa lidských zdrojů.
- **ERP Corporate Services** – řízení interních procesů. Obsahuje moduly Global Trade Services (GTS), Portfolio and Project Management (PPM), Quality Management (QA), Real Estate Management (REM), Enterprise Asset Management a další.
- **ERP Operations** – logistika.

Názvy řešení je třeba brát s rezervou. Často se stává, že pro marketingové účely se občas mění pojmenování nebo uspořádání řešení. Jejich výčet jsem uvedl za účelem objasnění vztahu mezi moduly a podnikovými scénáři.

### 2.3.3 Transakce

**Transakce** jsou úkony, které probíhají v rámci modulů. Provádějí je uživatelé při své každodenní práci se systémem. Každá transakce je krokem určitého procesu – např. zadání objednávky, vytvoření nového zaměstnance. Transakce však nemusí vždy probíhat pouze v rámci jednoho modulu. Některé procesy mohou způsobit spouštění transakcí přes různé moduly.

Obrázek 4 ukazuje vztahy mezi předchozími pojmy.

**Obrázek 4:**

*Vztahy mezi komponentami, moduly, podnikovými scénáři a transakcemi. Zdroj: upraveno dle (Anderson 2012, s. 34)*

### 2.3.4 Klienti

Dalším pojmem, který je třeba objasnit, je klient. „*Klienty se obvykle rozumí samostatné podniky či jednotky v rámci každého systému SAP*“ (Anderson 2012). Každý velký podnik má zpravidla mnoho divizí, poboček, nebo zastoupení v různých zemích. Systém klientů umožňuje pro každou takovou jednotku definovat zvláštní systém, který je součástí celkového systému SAP.

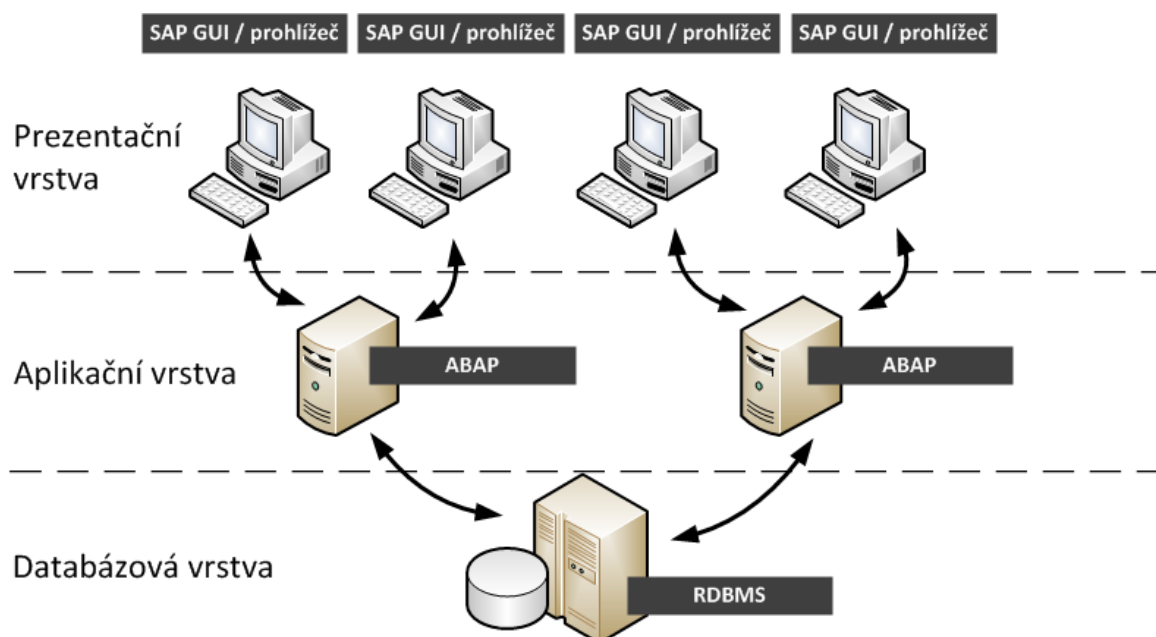
Klienty může podnik členit dle **dceřiných společností**, nebo na **geografickém základě**. Každý klient má pak svou vlastní nastavení parametrů a sadu kmenových dat. Je označen trojmístným číslem a je třeba ho zadat při přihlašování do systému.

V prostředí systému SAP je možné se setkat s dalším významem pojmu „*klient*“. A sice jako označení pro skupinu stavebních bloků, tzv. *building blocks*, platformy SAP NetWeaver – více o této problematice viz kapitolu 3.6.

## 2.4 Technologická architektura

Do této chvíle jsem popisoval systém SAP z hlediska uspořádání aplikací do komponent, modulů, nástrojů, samostatných jednotek apod., což se označuje jako **aplikační architektura** systému SAP (Koubek 2006, s. 17) nebo obecně softwarová architektura ERP systému (Gála et al. 2009, s. 160). Z **technologického hlediska** se systém SAP dělí do tří vrstev – prezentační vrstva, aplikační vrstva a databázová vrstva, jak ukazuje Obrázek 5. Toto rozdělení pak vytváří architekturu *klient/server*. (Kühnhauser 2012).





**Obrázek 5:**

*Schéma technologické architektury systému SAP. Zdroj: upraveno dle (Kühnhauser 2012, s. 17)*

**Prezentační vrstva** zajišťuje komunikaci uživatele se systémem a probíhá na klientské části. Prezentační vrstvu systému SAP tvoří prezentační servery a komunikují s aplikačním serverem prostřednictvím programového balíku SAP GUI.

**Aplikační vrstvu** tvoří SAP Application Server (aplikační server) a komunikuje s uživatelem přes rozhraní SAP GUI. Na aplikačním serveru se centrálně zpracovávají data, která se ukládají a načítají z databáze přes databázový systém.

**Databázovou vrstvu** tvoří databázový server, na kterém běží jedna společná databáze. Systém SAP používá nejčastěji databázi Oracle nebo MS SQL Server. Podle objemu dat mohou být databázové nebo aplikační servery umístěny na jednom nebo více počítačích.

Systém SAP musí být také provozován na nějaké kombinaci operačního systému a hardwaru. Konkrétní hardware, operační systém, databázový systém a další technologie nezbytné pro provoz systému SAP dohromady tvoří základ, který se nazývá *SAP Basis*. (Anderson 2012)

## 2.5 Odvětvová řešení

Každé odvětví má svá specifika (odlišné procesy, legislativní požadavky, řízení rizik). Proto systém SAP není vytvořen univerzálně pro každé odvětví, ale umožňuje přizpůsobit nasazený systém konkrétní oblasti, ve které podnik působí. K tomu se využívá tzv. **odvětvové řešení**, které se vždy instaluje „nad“ ostatními produkty SAP – např. nad SAP ERP. Tím se využívá jádra, které svou funkcionalitou pokrývá požadavky většiny typů podniků. Odvětvové řešení pak zajistí tu funkcionalitu, kterou vyžaduje konkrétní odvětví.

V literatuře – např. (Gála et al. 2009, s. 159) – se lze také setkat s označením „*branžové řešení*“.

Výhodou odvětvových řešení je nejen to, že usnadňují implementaci v různých oblastech podnikání, ale zejména to, že společnost SAP promítá do svých řešení nejlepší postupy (*best practices*) pro dané odvětví. Přijetím nejlepších postupů může podnik efektivněji uspokojovat potřeby svých zákazníků, činit lepší strategická rozhodnutí a zvyšovat výkonnost celého podniku.

Odvětvová řešení se dle (SAP 2013c) dělí do 4 oblastí: **Veřejný sektor**, **Finanční sektor**, **Výroba** a **Služby**. Každá oblast se pak dělí na další skupiny odvětví. Např. Výroba obsahuje skupiny Letectví & obrana, Automobilový průmysl, Chemický průmysl, Průmysl spotřebního zboží, Strojírenství & stavebnictví, Průmyslová elektronika, Průmyslová výroba & součásti, Přírodní vědy, Hutnictví, Petrochemický průmysl.

## 2.6 SAP a architektura orientovaná na služby

Ještě než přistoupím technologickému základu SAP NetWeaver, je třeba objasnit trend, který spočívá ve směřování k architektuře orientované na služby (SOA), což velmi souvisí s důvody vzniku SAP NetWeaver.

### 2.6.1 SOA

SOA (Service Oriented Architecture) definuje (Gála et al. 2009) jako: „*politiky, praktiky a rámce, které umožňují, aby funkcionality aplikací byla poskytována a spotřebována jako množina služeb, a to v takové úrovni granularity (rozsahu funkcionality), kterou potřebuje příjemce služby. Ten je oddělen od implementace služby a používá pouze jednoduché na standardech založené rozhraní*“. Oproti klasickým systémům, které jsou od základu navrhovány a vyvíjeny pro určité požadavky zákazníka, jsou systémy v SOA poskládány z mnoha znovupoužitelných služeb, které mezi sebou komunikují a realizují podnikové procesy.

Postupem času se ukázalo, že architektura orientovaná na služby je jedinou architekturou, která dokáže pružně reagovat na nové požadavky podniku na informační systém (Gála et al. 2009). Vysoké náklady při návrhu SOA jsou později mnohokrát pokryty díky úspoře nákladů při úpravách funkcionality podnikových aplikací. Není třeba vyvíjet nové aplikace, nebo upravovat ty současné při změně podnikových procesů. Stačí zajistit funkcionality

výměnou služeb nebo volbou jiných služeb s jinou úrovní granularity. Významné užití mají konkrétně tzv. webové služby, které tvoří základ pro *cloud computing*<sup>10</sup>.

Použitím SOA lze zajistit otevřenou a pružnou IT architekturu. Současným trendem je, že všichni nejvýznamnější poskytovatelé podnikových řešení připravují své aplikace pro architekturu orientovanou na služby, jak tvrdí (Anderson 2012), a společnost SAP v tomto směru není výjimkou.

## 2.6.2 Adaptační společnost SAP na SOA

Společnost SAP, stejně jako její konkurenti, promítá do svých produktů moderní technologie. V oblasti SOA zavedla pojem Enterprise SOA (podniková architektura SOA), kterým označuje pružnou, otevřenou IT architekturu, která umožňuje vytváření kompozitních aplikací na základě znovupoužitelných služeb.

Prvním krokem k fenoménu SOA bylo vytvoření Webového aplikačního serveru (Web Application Server), což v podstatě znamenalo rozšíření technologického základu systému SAP do webového prostředí. Cílem bylo zjednodušit instalaci a údržbu a přidat podporu pro jazyk Java. Postupem času se WAS stal platformou podporující bohaté využití webových technologií, jako jsou např. webové služby. Jeho součástí byla rozšířená podpora XML, protokolu SOAP a jazyka WSDL. To vše později vyústilo k nasazení platformy SAP NetWeaver, o které pojednává následující kapitola.

Mezi hlavní přínosy patří rychlost nasazení a prototypování aplikací, snadná údržba a s tím související nízké náklady na vývoj. Díky pružné architektuře jsou podniky schopné inovovat a propojovat své aplikace např. s dodavateli nebo odběrateli, což motivuje stále více firem k odklonu od nepružné architektury klient-server.

Společnost SAP dále definovala pět základních principů, díky kterým se stala leaderem na trhu s aplikacemi postavenými na základě SOA. Principy lze chápat i jako sadu pravidel pro vývoj těchto aplikací s podporou cloud-computingu (Anderson 2012, s. 327):

- **Abstrakce** – slouží ke skrytí podrobností
- **Modularita** – umožňuje vývoj znovupoužitelných komponent.
- **Standardizace připojení** – používá se k popisu a sdílení dat, které pak slouží k vytváření a kombinování služeb. Z těch je pak možné poskládat plně funkční podnikové procesy, scénáře a řešení.

---

<sup>10</sup> Cloud computing je použití výpočetních zdrojů (hardware a software), které jsou poskytovány jako služba přes internet (Anderson 2012, s. 315).

- **Volná vazba** – vlastnost, která umožňuje další vývoj dalších služeb bez nutnosti jejich přepisování. Zajišťuje opětovnou použitelnost a propojení mezi ostatními službami.
- **Přírůstkový návrh** – umožňuje měnit konfiguraci a skladbu služby bez nutnosti jejich opětovného sestavování.

Na závěr je nutné poznamenat, že Enterprise SOA od společnosti SAP je pouze přístupem k inovativnějšímu výpočetnímu modelu. Hlavní produkty, jako SAP ERP jsou samozřejmě stále založeny na architektuře klient-server, ale obsahují technologii SAP NetWeaver, která umožňuje propojovat jejich části s cizími aplikacemi a zejména webovými službami a tak mohou být součástí SOA. V tomto směru lze pozorovat trend společnosti SAP, který spočívá ve směřování do cloudu. Pro další informace o tom, jak společnost SAP promítá principy cloud computingu do svých produktů, doporučuji zdroj (Anderson 2012, s. 328) nebo aktuální webové stránky SAP<sup>11</sup>.

---

11 <http://www.sap.com>

## 3 SAP NetWeaver

V této kapitole charakterizují technologický základ SAP NetWeaver a jeho strategické využití.

Cílem kapitoly je vymezit pojem SAP NetWeaver a jaký je jeho vztah k systému SAP.

### 3.1 Důvody vzniku platformy SAP NetWeaver

**SAP NetWeaver** je **technologickým základem** naprosté většiny dnešních produktů firmy SAP a umožňuje jejich provoz a rozvoj. V tomto smyslu lze nahlížet na SAP NetWeaver jako na **horizontální podpěry**, na kterých stojí aplikace SAP, jak uvádí (Anderson 2012, s. 91). Ve smyslu integrace se pak hovoří o **vertikální podpoře** k propojení aplikací a rozšíření systému SAP do webového prostředí. V tomto kontextu se pak hovoří o SAP NetWeaver jako o integrační **platformě**. V této kapitole proto nazývám SAP NetWeaver platformou.

Ještě než charakterizují architekturu a možnosti požití SAP NetWeaver, uvádím na tomto místě stručný popis vzniku této technologie, jako reakci na:

- problémy spojené s integrací aplikací,
- tlak na snižování TCO,
- směřování k architektuře orientované na služby.

90. léta 20. století znamenala rozmach ERP systémů. Jak jsem uvedl v kapitole 2.1, základ podnikové aplikace se začal rozrůstat o další části. Kdykoliv se objevil nějaký požadavek podniku, vytvořila se nová aplikace. ERP systémy se rozrůstaly do všech oblastí – už nepomáhaly pouze plánovat podnikové zdroje, jak napovídá jeho zkratka, ale řídily např. dodavatelské řetězce, vztahy se zákazníky, vztahy s dodavateli apod. Pro každou funkcionalitu si podniky mohly koupit aplikaci. Problémem bylo, že i když měl ERP systém **jednu databázi** – což je jedna z jeho hlavních charakteristik – všechny dokoupené aplikace měly databáze **vlastní**. Vyhledávání souvisejících informací např. o jednom zákazníkovi ve třech systémech bylo náročné a začala se řešit **integrace**.

Integrace podnikových aplikací je však poměrně náročnou záležitostí. K propojení dvou systémů se používá tzv. **point-to-point** integrace, kdy je každá aplikace propojena s každou (Gála et al. 2009, s. 95). Propojené systémy však obsahovaly různorodá data. Jeden systém může uchovávat jména klientů jako jedno pole, druhý jako dvě pole (jméno a příjmení), jeden systém agreguje finanční data za čtvrtletí, druhý za měsíc apod. Kromě toho vzniká velké množství vazeb, což vede ke „špagetové integraci“ a jako řešení se mohou použít zprostředkující aplikace – tzv. **integrační brokery**, které transformují data z jedné aplikace pro potřeby té druhé. Transformace je možná pomocí API (*Application Program-*

*ming Interface*), kterým každá aplikace např. říká, v jakém formátu může přijímat data. API poskytuje pohled, jak je aplikace navržena, ale obsahuje přístup jen k některým částem programu. Problém API spočívá v tom, že i když programátor aplikaci rozumí, ne vždy může udělat to, co chce.

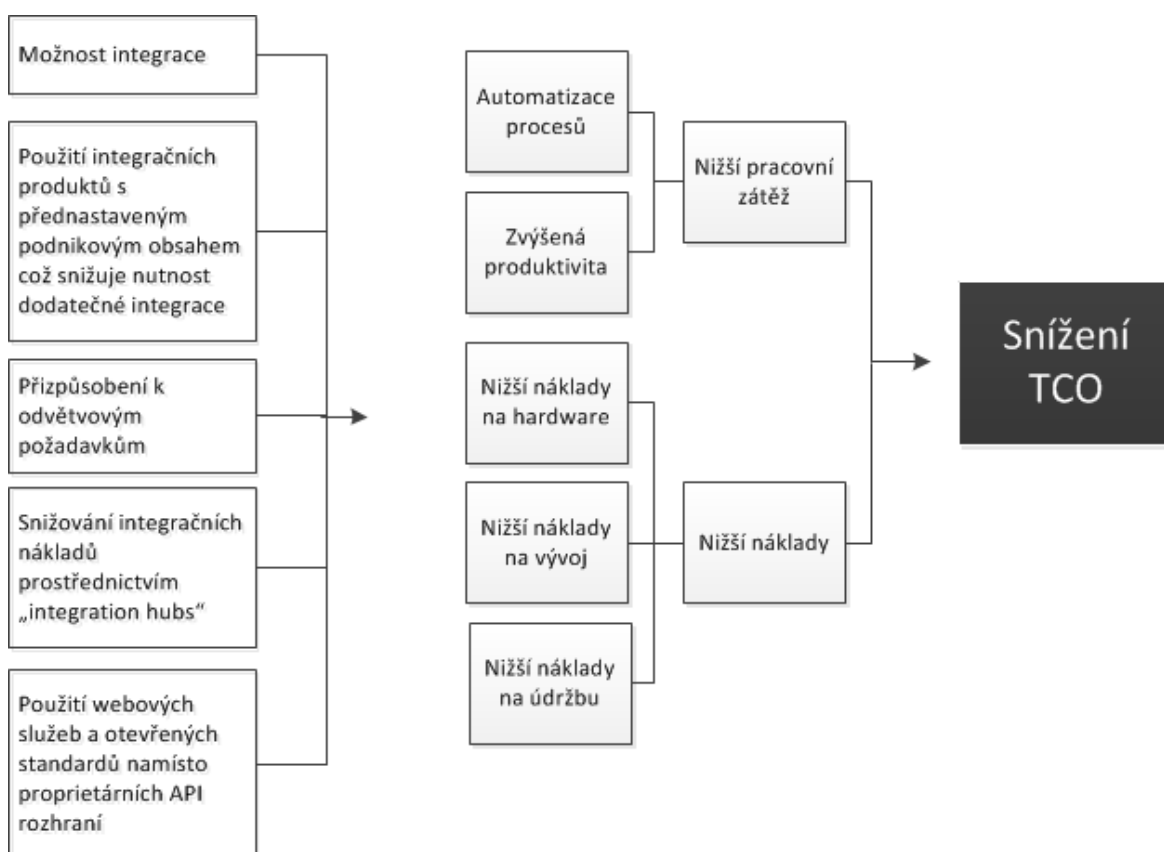
Největším problémem integrací však bylo (a současně je), že velmi zatěžují TCO (*Total Cost Of Ownership*). TCO jsou celkové náklady, které firma vynakládá už jen kvůli tomu, že podnikovou aplikaci vlastní – velkou položkou těchto nákladů je právě udržování point-to-point integrací. Proč je udržování integračních vazeb tak nákladné? Protože vyžadují mnoho schopných programátorů, kteří musí obstarávat úpravy rozhraní kdykoliv se aplikace změní.

Dle (Woods, Word 2004a, s. 33) z pohledu TOC zahrnuje:

- náklady na celou aplikaci (licence),
- náklady na integrační platformu,
- a náklady na integraci vlastních různorodých technologií.

Tyto náklady se nemusí řešit, pokud podnik koupí vše v jednom, jako jedinou integrační platformu, kterou je SAP NetWeaver.

SAP NetWeaver snižuje TCO mnoha způsoby, jak ukazuje Obrázek 6. Vytváří interoperabilitu mezi aplikacemi a platformami (odstínění problémů s point-to-point integrací), používá produkty upravené speciálně pro potřeby podniku a hlavně umožňuje vytvářet architekturu orientovanou na služby.



Obrázek 6:

*Schéma snižování TCO díky SAP NetWeaver. Zdroj: upraveno dle (Woods, Word 2004a, s. 35)*

Koncept SAP NetWeaver je založen na tzv. **předpřipravené integraci** (Woods, Word 2004a, s. 17). To znamená, že veškeré integrační záležitosti byly vyřešeny již při **tvorbě aplikace** a nemusí se dodatečně programovat po dokončení aplikace, jak to většinou bývá. Všechny aplikace postavené na platformě SAP NetWeaver mohou komunikovat mezi sebou i s aplikacemi „venku“. Tím se s trochou nadsázky dá říct, že je možné koupit si vyřešenou a fungující point-to-point **integraci jako produkt**. V praxi to ale znamená, že je možné si koupit sadu programových prostředků pro tvorbu point-to-point integrace, kde je již z 80 – 90 % práce hotová. SAP NetWeaver pro to poskytuje mnoho nástrojů, které budou popsány dále. Point-to-point integrace je samozřejmě úplně hotová pro celé řešení SAP Business Suite, ale funguje na otevřeném principu, takže je možné ji upravovat. Společnost SAP si uvědomuje, že v podnicích se mohou vyskytovat i jiné systémy, a tak její produkty zahrnují standardy, jako např. WSDL (*Web Services Description Language*), což utváří platformu pro tvorbu point-to-point integrace se systémy od jiných firem.

Úspory nákladů jsou realizovány také prostřednictvím **sdílených funkcí**. Vykazuje-li určitá část podnikové aplikace známky znovupoužitelnosti, tak se neprogramuje znovu, ale je v nezměněné podobě použita jinde v systému. Každá další verze platformy SAP NetWeaver obsahovala více a více sdíleného kódu, což později vedlo k myšlence, že jednotlivé části podnikové aplikace je možné poskytovat jako služby. Použit přístup přes API znamená být závislý na programovacím jazyce aplikace, kdežto služby jsou platformy samy o

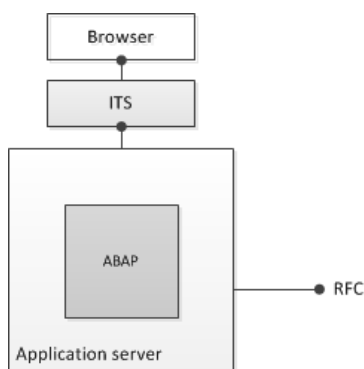
sobě a jazykově nezávislé brány k funkcionalitě aplikace. Díky SAP NetWeaver je funkcionalita řešení SAP Business Suite dostupná prostřednictvím služeb a podnik může získat všechny výhody architektury orientované na služby (SOA). Tím vznikla platforma SAP NetWeaver jako reakce na trend směřování k SOA.

## 3.2 Od webového aplikačního serveru k SAP NetWeaver

Pro pochopení role platformy SAP NetWeaver v rámci systému SAP, je třeba objasnit, jak tato technologie vznikala během vývoje technologické architektury systému SAP.

V kapitole 2.4 jsem popisoval, že prvkem technologické architektury systému SAP je aplikační server. Prvním aplikačním serverem systému SAP při nástupu éry klient/server byl *SAP Application Server*, který byl zároveň součástí technologického základu označovaného jako **SAP Basis** – viz Obrázek 7.

Uživatelé k němu přistupovali buď přes rozhraní SAP GUI nebo přes ITS (*Internet Transaction Server*). ITS nebyl platformě nezávislý, protože obsahoval služby, které byly úzce svázány s MS Windows. Celý aplikační server byl založen na jazyce **ABAP**, ve kterém byly vyvinuty také aplikace běžící na tomto serveru. Uživatelské rozhraní se tvořilo pouze pomocí modelovací metody zvané *Dynpro* nebo též *dynamic programming*. S ostatními aplikacemi komunikoval server prostřednictvím RFC (*Remote Function Call*) nebo speciálně vyvinutými konektory (Kangas 2005, s. 8). Tento stav reflektuje výše zmíněné **problémy s integrací**, protože konektory bylo jistě velmi nákladné vyvíjet a udržovat, protože byly speciálně navrženy pro každou aplikaci. Kdykoliv došlo ke změně připojeného systému, tak bylo třeba konektor předělat.



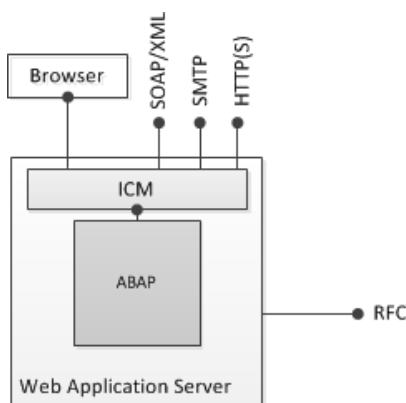
**Obrázek 7:**

*Schéma SAP Basis verze 4.6 a nižší. Zdroj: upraveno dle (Kangas 2005, s. 8)*

V 90. letech, v době internetového boomu, se začínaly aplikace přizpůsobovat webu. Výsledkem bylo uvedení webového aplikačního serveru označovaného zkratkou **WebAS** nebo WAS (*Web Application Server*). Tento server rozšířil technologický základ SAP Basis do webu, a to integrací serveru ITS, který byl původně samostatným externím produktem (Anderson 2012, s. 92). Z ITS vznikla komponenta ICM (*Internet Connection Manager*),



integrovaná do WebAS, která řídí komunikaci webového prohlížeče a WebAS stejně, jako je řízena komunikace se SAP GUI. ICM také podporoval komunikační protokoly (např. SOAP/XML), viz Obrázek 8.



**Obrázek 8:**

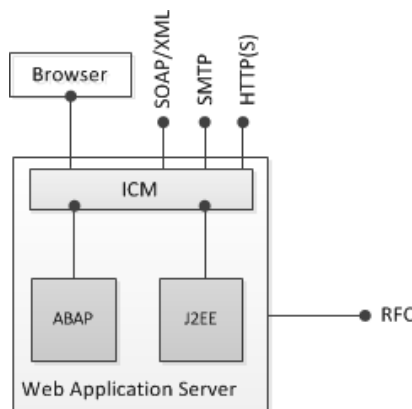
*Schéma Web Application serveru. Zdroj: upraveno dle (Kangas 2005, s. 8)*

V další verze WebAS byla již obohacena o prostředí Java. Tím bylo možné vyvíjet a nasa-zovat aplikace na WebAS jak v jazyce **ABAP**, tak v jazyce **Java**. Podpora jazyka Java výrazně rozšířila možnosti tvorby dynamických webových stránek jako uživatelského roz-hraní, a to prostřednictvím JSP (*Java Server Pages*). Další novinkou byla technologie Web Dynpro, která se dodnes užívá pro jednoduchou tvorbu webového uživatelského rozhraní (Kangas 2005, s. 9).

V kapitole 2.4 jsem dále uváděl rozdělení systému SAP na **technologickou** a **aplikační vrstvu**. Záměrem společnosti SAP při vzniku WebAS bylo také to, aby mohla poskytovat technologickou vrstvu **samostatně jako produkt** a nezávisle na podnikových aplikacích. Variantou byl

- tradiční technologický základ ABAP (*ABAP Stack*),
- technologický základ Java (*Java Stack*),
- nebo obojí.

Rychlý vývoj technologií vyžaduje pravidelné upgrady softwarových a hardwarových komponent, což způsobuje vysoké náklady a potíže s integritou aplikací (Voříšek 2011, s. 47). Díky oddělení technologické a podnikové vrstvy mohou podniky **provádět upgrady systému** SAP zvlášť pro každou vrstvu a ne najednou (Anderson 2012, s. 92). Schéma WebAS, rozšířeného o prostředí Java – viz Obrázek 9.



Obrázek 9:

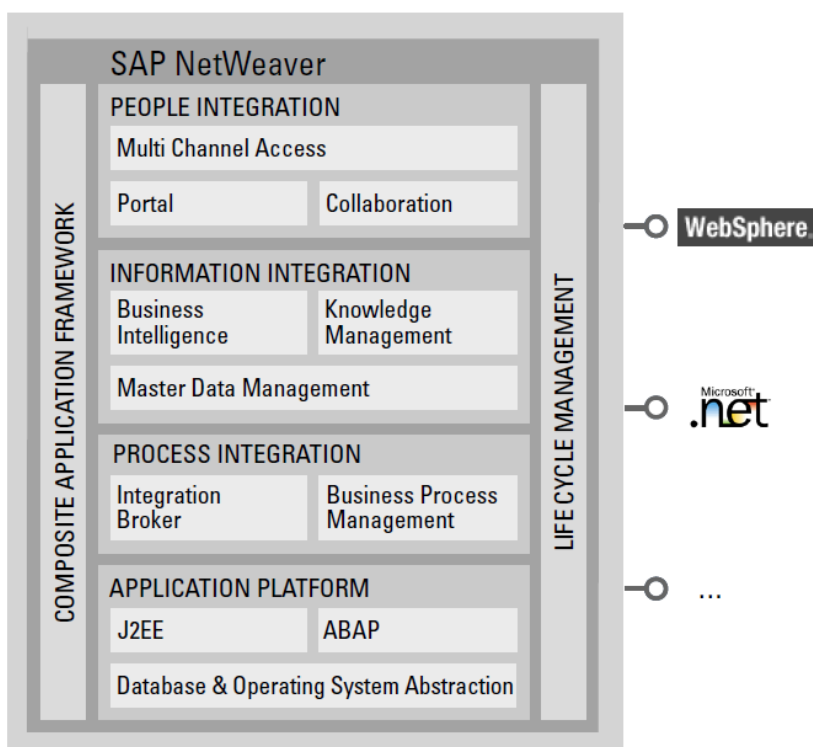
*Schéma WebAS rozšířené o prostředí Java. Zdroj: upraveno dle (Kangas 2005, s. 9)*

V roce 2004 byl koncept **technologického základu** rozšířen na celou **integrační** platformu. Tak **vznikla platforma SAP NetWeaver**, která měla za cíl zjednodušit připojení systému SAP k dalším podnikovým aplikacím. Do SAP NetWeaver byl nejprve přidán produkt *SAP Business Warehouse* a *SAP Business Enterprise Portal* (viz dále). Později byl přidán produkt *SAP NetWeaver Process Integration*, díky kterému se snáze propojují systémy ostatních výrobců se systémem SAP. Kdykoliv pak došlo k upgradu jakékoliv komponenty SAP Business Suite, bylo to příležitostí k integraci komponenty se SAP NetWeaver.

### 3.3 Schéma platformy SAP NetWeaver

Na tomto místě popisují základní schéma platformy SAP NetWeaver, které pochází z roku 2004. O té doby došlo k výraznému rozšíření a přejmenování komponent a nástrojů, základní princip však zůstává stejný – SAP NetWeaver řeší 3 úrovně integrace:

- **Integrace informací** – pomáhá seskupit informace z různých zdrojů a nahlížet na ně ve správném kontextu. Tím vytváří jednu verzi pravdy.
- **Integrace lidí** – dovoluje pracovníkům pracovat společně a efektivněji díky transparentnímu přístupu k funkcionalitě systému SAP a jednotnému pracovnímu prostředí.
- **Integrace procesů** – umožňuje koordinovat pracovní tok napříč odděleními, divizemi a samostatnými podniky a s využitím sdílených dat jednotlivých komponent systému SAP. Vytváří tak jednotný pohled koncových uživatelů na firemní procesy.



**Obrázek 10:**

*Schéma platformy SAP NetWeaver. Zdroj: (Woods, Word 2004a, s. 43)*

Obrázek 10 znázorňuje jednotlivé integrační úrovně. Všechny 3 úrovně stojí na aplikační platformě, kterou tvoří **Web Application server**, kterým jsem se zabýval v předchozí podkapitole. Schéma obsahuje ještě dva prvky, které jsou umístěny po stranách integračních oblastí, protože hrají důležitou podpůrnou roli v rámci celé platformy SAP NetWeaver. Jde o *Composite Application Framework* (CAF), který slouží k tvorbě kompozitních (skládatelných) aplikací (viz dále kap. 3.4.5). Díky CAF lze modelovat, kombinovat a stavět nové aplikace z již existujících aplikací, což je možné použít pro tvorbu architektury SOA. Druhým prvkem je *Lifecycle Management* – sada nástrojů pro bezproblémovou a rychlou instalaci, upgradu a opravy chyb aplikací.

Každý uvedený prvek schématu podporují nějaké komponenty platformy SAP NetWeaver, které popisují v následující části.

### 3.4 Oblasti SAP NetWeaver

Nyní se již zaměřím na konkrétní komponenty platformy SAP NetWeaver, které **pomáhají dosahovat výhod** uvedených v předchozích kapitolách. Celá platforma SAP NetWeaver obsahuje mnoho komponent, které představují aplikace a nástroje nebo i skupiny aplikací a nástrojů, které vyvinula společnost SAP. Od uvedení této platformy uběhlo již 9 let. Za tu dobu se velmi rozrostla a utvořila mnoho produktů, které (Anderson 2012) rozděluje do 6 kategorií (oblastí):

- Řízení základu
- Middleware
- Řízení informací
- Produktivita týmu
- Kompozice
- Řízení podnikových procesů

Pro srovnání uvádím, že v době vzniku platformy uváděla společnost SAP rozdělení pouze na dvě oblasti (Woods, Word 2004a):

- Integroční komponenty
- Vývojové a řídicí nástroje

Nyní z každé oblasti charakterizují nejdůležitější komponenty, jejich využití a jakým způsobem pomáhají dosahovat cílů platformy SAP NetWeaver.

### 3.4.1 Řízení základu (Foundation Management)

Řízení základu je označení pro komponenty, které jednak tvoří základ celé platformy a také zajišťují úspěšnou implementaci systému SAP. Patří sem následující komponenty:

- **SAP NetWeaver Application Server<sup>12</sup>** – základ, na kterém stojí celé řešení SAP Business Suite a také ostatní komponenty SAP NetWeaver. Jde prakticky o synonymum aplikačního serveru WAS, který jsem popisoval v souvislosti s evolucí celé platformy. Tato komponenta je úzce spjata se svým vývojovým prostředím, protože umožňuje programovat zákaznické aplikace a je zároveň běhovým prostředím, ve kterém jsou aplikace spouštěny. SAP NW AS má tři možnosti instalace podle toho, v jakém jazyce bude možné aplikace vyvíjet – základ ABAP, základ Java, nebo obojí. Zákaznickým vývojem v této komponentě se zabývá celá praktická část této práce – viz. kapitoly 3 a 4.
- **SAP NetWeaver Identity Management** – jedná se o nástroj pro správu uživatelských identit. Umožňuje koncovým uživatelům jednotný přístup ke všem aplikacím systému SAP. Více o problematice správy a řízení identit viz např. (Bertino, Takahashi 2010).
- **SAP NetWeaver Solution Manager (SolMan)** – tento nástroj používá společnost SAP pro řízení implementací systémů a jejich provozu. Díky nástroji SolMan je možné sledovat verze instalovaného kódu, oddělovat lokální kustomizace od jádra produktu a systematicky instalovat záplaty a upgrady systému SAP. Pomáhá také

---

<sup>12</sup> dále jen SAP NW AS

spravovat a konfigurovat komponenty SAP Business Suite, komponenty SAP NetWeaver a systémy od jiných dodavatelů na jednom místě. Tím pádem SolMan snižuje náklady na údržbu aplikací a zvyšuje produktivitu implementačního týmu.

### 3.4.2 Middleware

Pojem Middleware obecně označuje „*software zajišťující transparentnost umístění distribuovaných programových prostředků, pomocí něj lze pracovat s distribuovanými zdroji, jako kdyby byly lokální*“ (Gála et al. 2006). Pro systém SAP jsou to nástroje, které slouží k propojení různorodých systémů, aby pracovaly se sdílenými daty a podporovaly podnikové procesy běžícími „nad“ těmito systémy (Anderson 2012, s. 94). Zde je vidět ta „transparentnost“, která je zajištěna jednotným rozhraním pro práci s různými propojenými systémy.

Pro tuto oblast slouží produkt **SAP NetWeaver Process Integration**, dříve označován jako *SAP NetWeaver Exchange Infrastructure* – viz např. (Koubek 2006, s. 18). PI poskytuje framework na tvorbu adaptérů<sup>13</sup> pro jednotlivé aplikace, aby spolu mohly komunikovat jak komponenty SAP, tak ostatní systémy. Pro integraci s dalšími podnikovými aplikacemi od jiných dodavatelů jsou často potřeba adaptéry třetích stran – např. od firmy Seeburger<sup>14</sup> nebo Informatica<sup>15</sup> (Paltzer 2010, s. 11). Velkou výhodou je, že se nemusí přepisovat všechna rozhraní pokaždé, když se změní některá z aplikací. Stačí přepsat jediné rozhraní k PI a šetřit tak mnoho času a peněz. Adaptéry fungují na principu zasílání zpráv na vysoké úrovni zabezpečení a s garancí doručení. Zprávy lze posílat tam i zpět a synchronizovat tak databáze, integrovat procesy, vzdáleně volat funkce aplikací nebo vytvářet B2B (*business- to-business*) integrace – to vše v reálném čase (Woods, Word 2004a, s. 52).

Společnost SAP rozděluje adaptéry do 3 kategorií – **technické**, **aplikační** a adaptéry **odvětvových standardů**. Mohou být určeny pro SAP NetWeaver PI nebo starší SAP NetWeaver XI. Pro každou z těchto aplikací se adaptéry nazývají mírně odlišně, nicméně rozdělení zůstává v podstatě stejné. Úplný přehled adaptérů lze nalézt na webu *SAP Community Network*<sup>16</sup>.

---

<sup>13</sup> Adaptéry jsou programy, které propojují systém SAP se systémy od jiných dodavatelů.

<sup>14</sup> <http://www.seeburger.com>

<sup>15</sup> <http://www.informatica.com>

<sup>16</sup> <http://www.sdn.sap.com/irj/sdn/index?rid=/webcontent/uuid/70d8e250-e892-2b10-ab8a-d04433cf158b>

**Technické adaptéry** jsou specifické **pro určitou technologii** a protokol. Komunikace mezi systémy SAP a ostatními systémy je možná pomocí následujících technologií:

- **IDOC** (*Interchange DOCUMENT*) – standardní formát pro výměnu dokumentů.
- **RFC** (*Remote Function Call*) – standardní vzdálená volání funkcí systému SAP.
- **File/FTP(S)** – souborové systémy a FTP servery.
- **SOAP** (*Simple Object Access Protocol*) – protokol pro volání webových služeb.
- **JMS** (*Java Messaging Services*) – služby pro zasílání zpráv. Patří mezi tzv. *Message Oriented Middleware* (MOM).
- **JDBC** (*Java Database Connectivity*) – komunikace s relačními databázemi.
- **SMTP/POP3/IMAP** – komunikace s poštovními servery.
- **EDIFACT/ANSI X.12** – výměna elektronických dat.

Název adaptéru se skládá většinou z názvu technologie a přízviska „adapter“. Je dodáván buď společností SAP, nebo její partnerskou firmou.

Dále existují adaptéry **specifické pro konkrétní aplikace** ERP, CRM, SCM atd. Díky těmto adaptérům je možné se připojit k aplikacím, jako jsou např. MS Dynamics CRM, IBM Lotus Notes, Oracle PeopleSoft a dalším.

Kromě toho SAP NetWeaver PI podporuje i **adaptéry specifické pro odvětví** a sloužící k propojení podniků (B2B). Pro výměnu dat mezi podniky patřící do specifického odvětví využívají adaptéry SAP např. následující standardy:

- **HL7** (*Health Level Seven International*) – standard pro výměnu dat v odvětví zdravotní péče,
- **UCCnet** a **Transora** – standard pro výměnu dat v odvětví výroby spotřebního zboží,
- **SWIFT** – standard pro výměnu dat z oblasti finančních transakcí,
- a další.

### 3.4.3 Řízení informací (Information Management)

Oblast informačního managementu se stará o to, aby se všechny potřebné informace dostaly ke správným lidem ve správný čas, a tím urychluje proces rozhodování v podniku. V této oblasti obsahuje SAP NetWeaver 4 produkty:

- **SAP NetWeaver Master Data Management** (správa kmenových dat) – systém pro harmonizaci informací, které jsou rozmístěny napříč mnoha aplikacemi. Master data (kmenová data) je označení pro data, která nejsou svázána s žádnou konkrétní transakcí (např. data o zákaznících, produktech, zaměstnancích) a nachází se v datovém skladu. V datovém skladu jsou všechny informace kopírovány do

centrálního úložiště a MDM umožňuje provádět operace, jako je konsolidace obsahu nebo harmonizace dat (Woods, Word 2004a, s. 51). To znamená, že data z distribuovaných úložišť budou konzistentní – např. nová adresa zákazníka se promítne do všech systémů. MDM také nemusí pracovat pouze s datovými sklady – např. v případě různých akvizic pomáhá podnikům sjednotit jejich data uložená v různých databázích (Anderson 2009, s. 95).

- **SAP NetWeaver Business Warehouse** – datový sklad od společnosti SAP a nástroje pro analýzy nad
- **SAP NetWeaver Business Warehouse Accelerator** – škálovatelné řešení, které slouží pro urychlení zpracování dotazů v datovém skladu.
- **SAP NetWeaver Information Lifecycle Management** (řízení životního cyklu informací) – tento produkt pomáhá získat úplnou kontrolu nad podnikovými daty pomocí automatizací procesů, které jsou spojeny s pořízením, uchováním, archivací a skartací dat. Je také zárukou toho, že podnik bude splňovat interní i externí regulace v této oblasti.

Celkově pomáhají produkty z oblasti informačního managementu snižovat náklady tím, že **předcházejí vzniku chyb** v datech.

### 3.4.4 Produktivita týmu (Team Productivity)

Tento koncept zahrnuje nástroje pro zvýšení efektivity koncových uživatelů, jednotlivců i celých týmů.

- **SAP NetWeaver Portal** – portálové řešení od společnosti SAP. Portál je „*množina technologií a aplikací tvořící univerzální rozhraní, jehož prostřednictvím je každému, koho se dotýkají činnosti organizace (zákazník, dodavatel, zaměstnanec, apod.), umožněno účastnit se procesů organizace, přistupovat ke všem relevantním informacím, komunikovat se všemi kooperujícími pracovníky a realizovat adekvátní aktivity spojené s podnikovými procesy*“ (Gála et al. 2009, s. 138). Portály např. řeší zvyšující se neefektivitu zaměstnanců, kteří musí obsluhovat několik různorodých systémů na jednu, které mají odlišný vzhled, způsob přihlašování a mnoho nepotřebných funkcí. Portály se také používají k tzv. *integraci na úrovni přístupu*, jak dále uvádí (Gála et al. 2009, s. 376). SAP NetWeaver Portal představuje **centrální přístupové místo** ke všem relevantním funkcím pro daného uživatele na základě propracovaného systému oprávnění (*role-based interface*). K úsporám nákladů dochází zejména díky **zvýšené produktivitě** zaměstnanců, kteří nemusí přepínat mezi aplikacemi, pamatovat si několik přístupových údajů a vyznat se v rozhraní, které má jednotný vzhled i když ve skutečnosti pracují s několika systémy i třeba od různých dodavatelů. Díky **definovaným rolím** má uživatel rychlý přístup přesně k těm funkcím a informacím, které potřebuje a nepotřebné funkce si může skrýt (Woods,

Word 2004a, s. 45). Další výhodu uvádí (Anderson 2012, s. 96), a ta spočívá v tom, že při použití portálového řešení si zaměstnanci nemusí na své počítače instalovat a dále udržovat tlusté klienty systému SAP.

- **SAP NetWeaver Mobile** – vývojové prostředí, které umožňuje rozšířit funkcionalitu aplikací i do oblasti mobilních zařízení. Pracovníci, kteří jsou často na cestách, se tak mohou **účastnit podnikových procesů a nemusí být nutně připojeni** k podnikové síti. Kdykoliv se pak k podnikové síti připojí, jejich data se synchronizují se serverem. Hlavní výhodou mobilního přístupu je zpřístupnění klíčových informací kdykoliv a kdekoliv. Dále pomáhá lépe integrovat procesy tím, že snižuje opoždění při získávání distribuovaných informací. SAP NetWeaver Mobile velmi **usnadňuje práci vývojářům**, kteří nemusí psát rozhraní aplikací speciálně pro každé zařízení (mobilní telefony, tablety). Stačí použít jediné jednotné rozhraní, které zajistí, že aplikace bude komunikovat s kterýmkoliv zařízením nyní i v budoucnu.
- **SAP NetWeaver Enterprise Search** – nástroj, který usnadňuje práci uživatelům vyhledávání informací napříč celým systémem i v nestrukturovaných dokumentech. Stejně jako SAP NetWeaver Mobile přispívá k urychlení procesu rozhodování a větší informovanosti uživatelů.

### 3.4.5 Kompozice (Composition)

Tato oblast obsahuje sadu nástrojů sloužících k propojování kompozitních aplikací, které pak dohromady pokrývají určité podnikové scénáře (Anderson 2012, s. 96). Jsou to aplikace složené z dalších hotových aplikací a mohou být součástí architektury orientované na službu.

- **SAP NetWeaver Composition Environment** – platforma pro tvorbu a běh kompozitních aplikací. Nabízí mnoho nástrojů pro tvorbu těchto aplikací v jazyce Java a jejich propojování jak s aplikacemi systému SAP, tak s jinými aplikacemi založenými na architektuře SOA.
- **SAP NetWeaver Developer Studio** – vývojové prostředí založené na *Eclipse*, které patří do kategorie open source. Zatímco Composition Environment se používá převážně k **propojování aplikací**, tak Developer Studio slouží pro **vývoj samostatných aplikací** založených na specifikaci Java EE a webových službách (Ferguson 2003). Obsahuje veškerou podporu pro editaci kódu, spouštění a ladění aplikací. SAP NetWeaver Developer Studio obsahuje vývojový nástroj a běhové prostředí pro tvorbu aplikací s webovým uživatelským rozhraním zvaný Web Dynpro. Díky Web Dynpro stačí uživatelské rozhraní pouze „*nakreslit*“ a specifikovat, jak se má chovat. Kód pro vytvoření rozhraní se generuje automaticky. Tento nástroj používají programátoři při vývoji dle architektury MVC (*Model View Controller*). Ke snižování nákladů zde dochází zejména šetřením času programátorů a zvyšováním jejich produktivity.



- **SAP NetWeaver Visual Composer** – grafické modelovací prostředí, které se používá pro tvorbu uživatelského rozhraní. Oproti nástroji Web Dynpro je Visual Composer zaměřen i na nevývojáře, protože je to nástroj čistě modelovací a nekombinuje se s psaním kódu. Často se používá k tvorbě tzv. *iViews*<sup>17</sup> pro SAP NetWeaver Portal.

### 3.4.6 Řízení podnikových procesů (Business Process Management)

SAP NetWeaver Business Process Management lze chápat jako podmnožinu **SAP NetWeaver Composition Environment**. BPM obecně je soubor nástrojů, jejichž prostřednictvím je možné podnikové procesy modelovat, spustit a realizovat a také monitorovat jejich průběh (Kunstová 2009, s. 80). V nástrojích společnost SAP se nejprve vytvoří definice a kroky jednotlivých procesů a nastaví se podniková pravidla a výjimky. Poté se modeluje průběh procesu pomocí standardní notace. Monitorování procesů pak pomáhá zlepšit jejich rychlost a efektivitu.

Pro modelování se používá notace BPMN (*Business Process Modeling Notation*) a je „*standardem pro grafickou reprezentaci firemních procesů v diagramech. Jeho doplňkem je Business Process Modeling Language (BPML), jazyk pro modelování a popis procesů vycházející z XML*“ (Řepa 2007, s. 125). Notace BPMN pomáhá snadno a intuitivně modelovat podnikové procesy a představuje společné prostředí jak pro byznys analytiky, tak pro vývojáře. Všechnu tuto funkcionalitu nabízí následující nástroje a jejich schéma ukazuje Obrázek 11.

- **Process Composer** – nástroj integrovaný do prostředí SAP NetWeaver Developer Studio. Zahrnuje dvě perspektivy – *Procesní modelování* (Process Modeling) sloužící k návrhu procesů a určené zejména byznys analytikům, a *Vývoj procesů* (Process Development), jejímž cílem je vyvinout procesy aby byly spustitelné, a je určena převážně vývojářům. Process Composer podporuje všechny fáze procesního modelování. Od obecného návrhu, přes vývoj až k nasazení a spuštění. Aspekty životního cyklu (verzování, transport<sup>18</sup>, atd.) jsou plně podporovány napříč SAP NetWeaver Composition Environment. SAP rozděluje v notaci aktivity na *lidské* (*human*) a *automatizované* (*automated*) aktivity. Tyto aktivity mohou být dále rozděleny na vnořené procesy (podprocesy). **Lidské aktivity** spouští úlohy, které vykonávají koncoví uživatelé, zatímco **automatizované aktivity** jsou vykonávány

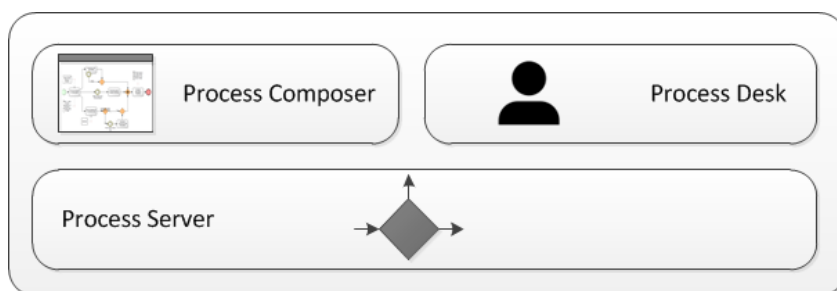
---

<sup>17</sup> iView představuje vizuálně vytvořený prostor na webové stránce a může obsahovat různá data, informace i služby.

<sup>18</sup> *transport* je označení pro mechanismy, které zajišťují přechod z vývojového prostředí do testovacího a nakonec do produkčního prostředí. Více o transporthách popisují v další kapitole.

webovými službami. Tím je možné je plně integrovat se SAP ESA (*SAP Enterprise Service Architecture*), což je architektura orientovaná na služby od společnosti SAP.

- **Process Server** – snadno použitelný nástroj, který je integrován do SAP NetWeaver Composition Environment a slouží k samotnému vykonávání modelů podnikových procesů. Představuje běhové prostředí procesů, kde je možné procesy spouštět, zastavovat a spravovat.
- **Process Desk** – představuje uživatelskou interakci s hotovými procesy a úlohami. Koncoví uživatelé používají Process Desk jako centrální přístupové místo, kde provádějí jednotlivé kroky podnikových procesů.



**Obrázek 11:**

*Schéma nástrojů SAP NetWeaver pro BPM. Zdroj: upraveno dle (SAP 2013b)*

### 3.5 Případy užití platformy SAP NetWeaver

Příručka *SAP NetWeaver Master Guide* (SAP 2010), která poskytuje „best practices“ pro implementaci platformy SAP NetWeaver, rozděluje komponenty do 6 kategorií, které označuje jako případy užití (*Use Cases*). Ty se v praxi definují podnikem, jako požadavky na platformu SAP NetWeaver a implementační tým musí zajistit jejich pokrytí komponentami.

1. Tvorba kompozitních aplikací (*Building Composite Applications*)
2. Použití datových skladů (*Data Warehousing*)
3. Tvorba integračních scénářů (*Building Integration Scenarios*)
4. Integrující a samoobslužný obsah (*Integrating and Self-servicing content*)
5. Mobilní přístup k podnikovým procesům (*Mobilizing Business Processes*)
6. Zákaznický vývoj (*Custom Development*)

### 3.6 Stavební bloky platformy SAP NetWeaver

Všechny komponenty SAP NetWeaver, které jsem uvedl v této kapitole, pokud by byly použity samostatně, nenabízejí podnikům takovou flexibilitu, jakou dnešní podniky vyža-

dují. Proto společnost SAP vyvinula koncept **instalovatelných softwarových jednotek**, které tvoří jakési základní kameny systémů SAP, a které (Anderson 2012) označuje jako tzv. *building blocks*. Ty pak pokrývají jednotlivé případy užití, uvedené výše. Instalují se jako povinné nebo volitelné jednotky v závislosti na IT architektuře podniku. Tyto bloky nebo jednotky mohou být dle (Anderson 2012, s. 98) rozděleny na tři základní druhy – druhy užití<sup>19</sup> (*Usage Types*), samostatné stroje (*Standalone Engines*) a klienti (*Clients*).

### 3.6.1 Druhy užití (Usage Types)

Druhy užití jsou softwarové jednotky, které tvoří základní stavební bloky platformy SAP NetWeaver. Jsou identifikovány jedinečným číslem (SAP system ID) a jsou nakonfigurovány ke specifickému účelu. Mezi tyto softwarové jednotky patří:

- *Development Infrastructure* (DI)
- *Enterprise Portal* (EP)
- *Enterprise Portal Core* (EPC)
- *Business Warehouse* (BW)
- *Process Integration* (PI)
- *Application Server Java* (AS Java)
- *Application Server ABAP* (AS ABAP)
- *Application Server Java* (AS Java)

Např. komponenta SAP NetWeaver Portal, o které jsem se zmiňoval v předchozí části, se skládá ze dvou jednotek – **Enterprise Portal Core** (obsahující základní komponenty, které se dříve instalovaly jako standardní instalace portálu) a **Enterprise Portal** (sdružující původní doplňkové komponenty, jako např. *Collaboration* nebo *Knowledge Management*)

### 3.6.2 Samostatné stroje (Standalone Engines)

Samostatné stroje jsou takové aplikace nebo služby, které se obecně kombinují s druhy užití a rozšiřují tak jejich funkcionalitu. Zde jsou jejich příklady:

- *Content Server*
- *Gateway*

---

<sup>19</sup> překlad „druhy užití“ z anglického pojmu Usage Types není úplně nejšťastnější. Nicméně je to jediný překlad, který se vyskytuje v literatuře (Anderson 2012), a proto se ho budu držet.

- *Search and Classification (TREX)*
- *Web Dispatcher*

Např. stroj pro vyhledávání a klasifikaci (*Search and Classification*) spolupracuje jak se SAP NetWeaver Portal, tak i se SAP Business Intelligence a slouží ke zvyšování výkonu jejich dotazovacích a vyhledávacích funkcí. SAP Web Dispatcher lze použít u systémů založených na WebAS ABAP i WebAS Java (Anderson 2012, s. 99).

### 3.6.3 Klienti (Clients)

Klienti jsou kombinace prezentačních komponent (např. SAP GUI, webový prohlížeč) a vývojových nástrojů zaměřených na úpravy uživatelského rozhraní. Zde jsou jejich příklady:

- *SAP GUI*
- *Business Explorer* (pro SAP BW)
- *Developer Workplace*
- *Developer Studio*
- *Adobe LiveCycle Designer*
- webový prohlížeč
- *Mobile Infrastructure Client*

### 3.6.4 Mapování stavebních bloků na případy užití

Při implementaci platformy SAP NetWeaver je třeba určit, které softwarové jednotky ze stavebních bloků nainstalovat, aby byly plně **pokryty požadavky** definované podnikem – tj. případy užití (viz kap. 3.5). Pro urychlení implementace poskytuje průvodce SAP NetWeaver Master Guide návod (SAP 2010), kde jsou pro každý případ užití vypsány softwarové jednotky, které je nezbytné nainstalovat. Chce-li podnik například v budoucnu rozšiřovat funkcionalitu systému SAP pomocí zákaznického vývoje (*Custom development*), musí jeho implementační tým nainstalovat alespoň funkční jednotku Application Server ABAP a klienta SAP GUI. Tento a další příklady mapování ukazuje následující tabulka.

**Tabulka 4:** Mapování softwarových jednotek na případy užití. Zdroj: (SAP 2010)

Případ užití	Stavební bloky platformy SAP NetWeaver		
	Druhy užití	Samostatné stroje	Klienti
<i>Tvorba kompozitních aplikací:</i> Vývoj v Javě a SOA infrastruktura	AS Java <i>Adobe Document Services</i> (volitelné) <i>Business Process Management a Business Rule Management</i> <i>Composite Application Framework</i> <i>Composition Environment Platform</i> <i>NW Product Description</i> DI (volitelné)		<i>Developer Workplace</i> <i>SAP NetWeaver Developer Studio</i> <i>Visual Composer</i>
<i>Tvorba kompozitních aplikací:</i> <i>Business Rules Management a Process Composition</i>	AS Java <i>NW Product Description</i> DI (volitelné) <i>Enterprise Services Repository</i>		Developer Workplace SAP NetWeaver Developer Studio Webový prohlížeč
<i>Využití datových skladů</i>	AS ABAP BW ABAP AS Java <i>NW Product Description Extensions</i> BI Java EP Core – Application Portal Enterprise Portal	<i>Search and Classification (TREX)</i>	SAP GUI s <i>BW Add-On</i> nebo <i>BI Add-On</i> <i>SAP Business Objects Advanced Analysis</i> pro MS Office (volitelné) <i>SAP Business Objects Crystal Reports</i> (volitelné) <i>SAP Business Objects Xcelsius</i> (volitelné) <i>SAP Business Objects Web Intelligence</i>
<i>Zákaznický vývoj</i>	Application Server ABAP		SAP GUI

### 3.7 Shrnutí přínosů platformy SAP NetWeaver

V celé této kapitole jsem popisoval komponenty SAP NetWeaver ve smyslu jejich vlastností a schopností řešit podnikové požadavky na informační technologie. Na několika mís-

tech jsem také zmínil, jak daná komponenta může šetřit náklady. V této podkapitole shrnuji ty **hlavní efekty**, které může SAP NetWeaver podniku přinést, protože nakonec nezáleží na tom, jak je technologie dokonalá, když nedokáže podniku vydělat nebo uspořít peníze a zajistit náskok v dnešním hyperkonkurenčním prostředí.

Z **marketingového hlediska** existuje mnoho strategií, jak získat konkurenční výhodu. Dle strategie „*modrého oceánu*“, představené ve stejnojmenné knize od autorské dvojice (Kim, Mauborgne 2005), by měly firmy přestat s konkurencí soupeřit a vyřadit ji ze hry tím, že vytvoří naprosto nový tržní prostor (modrý oceán). V některých případech toho lze dosáhnout díky pokrokové inovaci, jako to kdysi udělala např. firma Sony se svým přenosným přehrávačem „walkman“ nebo firma Apple, která jako první založila vlastní trh stolních počítačů. Všechny inovace by měly být obtížně napodobitelné, aby tento nový trh zůstal alespoň nějakou dobu „modrý“. Později se však začnou na trh pronikat další a další firmy a oceán se začne měnit na „rudý“, který tvoří všechna dnes existující odvětví.

Je pochopitelné, že většina dnešních firem soupeří navzájem právě v rudých oceánech. Nesnaží se vymýšlet patenty na nové výrobky (které konkurence stejně rychle napodobí), ale raději se snaží porazit konkurenci na trhu již existujících výrobků např. strategií nízkých nákladů, což je jedna ze základních Porterových strategií (Kotler, Keller 2007, s. 97), a mohou tak nabídnout zákazníkům výrobky s nižší cenou. Příkladem je firma Dell, která nevynalezla počítač ani žádnou z jeho částí, ale dokázala inovovat podnikové procesy a vynalezla lepší a levnější způsob, jak sestavovat počítače a dostat je do rukou zákazníků. Prvenství v nákladech lze získat **zdokonalováním podnikových procesů**, které budou vykonávány rychleji a efektivněji, než konkurence. Zdokonalení procesů, jako např. zákaznický servis, výroba a zásobování je jistě udržitelnější konkurenční výhoda, než krátkodobý náskok díky vytvoření nové poptávky, jak uvádí také (Woods, Word 2004a, s. 62).

**Dokonalejších procesů** lze dosáhnout různými technologiemi, které se rychle vyvíjejí, jsou dnes čím dál levnější a informace čím dál dostupnější, což uvádí (Voříšek 2011, s. 78) jako jeden z trendů ICT. Dostupnost informací se také značně projevuje ve zvyšování vlivu nástrojů typu *Competitive Intelligence* (konkurenční zpravodajství).

Podniky proto rychle přijdou na to, jak pomocí technologií a informací napodobit konkurenční „*best practices*“. Avšak skutečnou udržitelnou konkurenční výhodou v tomto prostředí, je dle (Řepa 2007) schopnost **přizpůsobovat podnikové procesy** tak rychle, jak se **mění podmínky**, ve kterých se firmy nacházejí. Firmy dobře vědí, co musí udělat pro změnu procesů, avšak nepružná a nákladná povaha jejich IT je od této změny odrazuje. Vysoké náklady jsou způsobeny investicemi do nepružných aktiv, jako je hardware a software, které nelze škálovat (musí být dimenzovány na maximální zatížení a v době snížených požadavků jsou zdroje nevyužity). Navíc se tato aktiva dlouho odepisují, což „*neumožňuje pružně a efektivně přizpůsobovat informační strategie změnám v hospodářském prostředí. ICT se tak může paradoxně stát i brzdou rychlého inovativního vývoje firmy*“ (Voříšek 2011, s. 48).

Pro tyto případy stanovila společnost SAP **teorii**, jak mají být **aplikace postaveny**, aby tvořili **pružnou architekturu orientovanou na služby** (SAP ESA) včetně využití webových služeb dle modelu SaaS. Jak už jsem uváděl dříve – viz kapitolu 2.6 – SAP NetWeaver znásobuje sílu aplikací SAP Business Suite, protože aplikace díky ní mohou být součástí této architektury.

**Přínosy SAP NetWeaver** mohou být rozděleny podle toho, na koho jsou zaměřeny v rámci rolí v podniku (Woods, Word 2004a, s. 63):

- **CEO** (*Chief Executive Officer*) – SAP NetWeaver umožňuje pružnou realizaci podnikové strategie, a tak se může podnik rychle přizpůsobovat měnícím se podmínkám.
- **IT Profesionál** – SAP NetWeaver zavádí inovaci do podnikových procesů napříč celým podnikem tím, že dokáže využít současnou IT infrastrukturu a přeměnit ji na rychle přizpůsobitelnou. SAP NetWeaver je také postaven tak, aby byl vývoj jednoduchý a tím byly změny levnější a méně riskantní.
- **CIO** (*Chief Information Officer*) – Ředitelé informatiky používají SAP NetWeaver jako platformu při rozhodování, jak docílit sladění IT a byznysu (*business-IT alignment*). Jak bylo zmíněno v úvodu této části, TCO zahrnují náklady na licence aplikací, integrační nástroje a integraci toho všeho dohromady. SAP NetWeaver snižuje TCO díky předpřipravené integraci a pomáhá snižovat celkovou komplexitu systémů.
- **Vývojář** – SAP NetWeaver jedna z nejpokročilejších platforem pro vývoj podnikových aplikací ve smyslu vývoje podřízeného modelu (*Model Driven Development*), řízení podnikových procesů (*Business Process Management*) a abstrakce na všech úrovních.

## 4 Zákaznický vývoj v jazyce ABAP

V této kapitole charakterizují pojem zákaznický vývoj a principy tvorby aplikací v jazyce ABAP.

Cílem je popsat vývoj v jazyce ABAP a jeho nezbytné nástroje alespoň do takové míry detailu, aby byl čtenář schopen vytvořit takové aplikace, které se běžně při zákaznickém vývoji programují.

### 4.1 Zákaznický vývoj

Ačkoliv je nabídka podnikových aplikací společnosti SAP považována za nejobsáhlejší na trhu, nemůže pokrýt všechny specifické požadavky zákazníků. Dokonce i přes to, že aplikace jsou schopny pružně reagovat na měnící se podnikové scénáře díky propracované konfiguraci, mohou nastat situace, kdy je třeba upravit jejich funkcionalitu, snadno je propojit s jinými aplikacemi nebo poskytnout jejich část pro architekturu SOA (SAP, 2013b). Realizace těchto úprav se souhrnně označuje jako zákaznický vývoj (*Custom development*).

Zákaznický vývoj je součástí celé implementace systému SAP, která se řídí dle metodiky ASAP<sup>20</sup> (*Accelerated SAP*). Ta z pohledu vývojáře zahrnuje také konfiguraci systému, která se provádí pomocí *konfiguračních nástrojů* (příručka *Implementation Guide*, IMG). Zákaznický vývoj se však realizuje pomocí *programovacích nástrojů*, které lze dále rozdělit podle toho, v jakém jazyce nebo prostředí vývoj probíhá:

- v jazyce ABAP,
- v jazyce JAVA,
- SAP NetWeaver Composition Environment.

Zákaznický vývoj ale nemusí při implementaci systému probíhat vždy. Dle (Anderson 2012, s. 275) probíhá pouze v případě potřeby:

- Vývoje takové funkcionality, která není součástí standardních programů SAP
- Vytváření speciálních výkazů vyžadovaných vedoucími pracovníky podniku a sloužící k podpoře řízení podniku

---

<sup>20</sup> Metodika ASAP zahrnuje postupy a doporučení při řízení projektu implementace systému SAP. Pomáhá projektovému týmu při optimalizaci času, lidí, a dalších prostředků nezbytných pro implementaci systému SAP (Anderson 2009, s. 250). Skládá se z pěti fází – *příprava projektu, plán podniku, realizace, konečná příprava a přechod do produkčního provozu*.



- Vytváření speciálních formulářů potřebných pro zadávání nových dat do systému (a to takovým způsobem, který zajistí vyloučení většiny rizik souvisejících se zadáváním dat lidmi)
- Propojení systému SAP s nějakým jiným systémem
- Vytvoření speciálních konverzních programů, umožňujících přenášení dat z jednoho systému do druhého

V následujících částech krátce popisují tři zmíněné jazyky (prostředí) a jejich nástroje, pomocí kterých zákaznický vývoj probíhá.

#### 4.1.1 ABAP

Jazyk ABAP (*Advanced Business Application Programming*) vznikl v 80. letech minulého století a vyvinula ho společnost SAP původně pro své databázové aplikace orientované na dialogy. Z obecného procesoru na formátování zpráv se postupně stal jazyk pro tvorbu podnikových aplikací.

Jazyk ABAP, navzdory svému stáří, stále zůstává hlavním jazykem pro programování v systému SAP (je v něm naprogramována většina samotného systému SAP). Vývoj probíhá v komponentě SAP NetWeaver Application Server a základním nástrojem je *ABAP Development Workbench*. Nabízí bohatou sadu funkcí, které vývojáři mohou použít k tvorbě nebo úpravám systému SAP. Další, dílčí nástroje jsou uvedeny dále, v celé této kapitole.

#### 4.1.2 Java

Vývoj v jazyce Java pro aplikační server SAP NetWeaver je založen na standardu Java EE. Společnost SAP zvolila tento jazyk kvůli jeho nezávislosti na platformě a chtěla také umožnit vývoj aplikací v systému SAP pomocí běžnějšího jazyka, než je ABAP. Jazyk ABAP měl být dokonce v jednu dobu úplně nahrazen jazykem Java, kdyby nedošlo k akvizici společnosti Sun Microsystems firmou Oracle v roce 2010, což je jeden z hlavních konkurentů firmy SAP (Anderson 2012, s. 277).

Pro vývoj v jazyce Java slouží SAP NetWeaver Developer Studio (NWDS), který vychází z open source nástroje Eclipse. NWDS obsahuje tyto funkce:

- Podpora standardu Java EE
- Podpora webových služeb
- Ladění aplikací za běhu
- Možnost nasazení či instalace za běhu
- Průvodce a grafické nástroje pro urychlení vývoje

Distribuce změn programů lze provádět stejně, jako při vývoji v jazyce ABAP pomocí Transport Management System (TMS) – viz dále.

### 4.1.3 Composition Environment

SAP NetWeaver Composition Environment je vývojové prostředí založené na standardech a orientované na služby. Slouží k vývoji aplikací, složených ze znovupoužitelných komponent společnosti SAP nebo třetích stran. Tím se výrazně urychluje inovace podnikových procesů.

## 4.2 Proces zákaznického vývoje

Zákazník nejprve nadefinuje své požadavky v dokumentu **RFP** (*Request For Proposal*). Ten obsahuje všechny informace o tom, co by měla nová aplikace nebo jakákoliv jiná úprava systému SAP řešit. V dokumentu může zákazník např. požadovat vyřešit systém objednávek, který je v jeho podnikání velmi specifický. Společnost SAP nebo její partner (dále jen dodavatel) odpoví dokumentem, který se nazývá **Blueprint**. Zde jsou popsány všechny procesy, které bude úprava řešit. Pokud zákazník dokument odsouhlasí, sestaví dodavatel další dokument, který obsahuje **funkční specifikace**, tedy seznam všech funkcí aplikace. Např. seznam obrazovek a návrhu jejich elementů. Po odsouhlasení funkční specifikace zákazníkem probíhá **vývoj a implementace** ve vývojovém systému SAP. Zákazník poté testuje aplikaci v systému pro zajištění kvality (konsolidačním systému, viz dále) a pokud je spokojen, tak podepíše tzv. **akceptační testy**. Dodavatel pak dodá technickou specifikaci aplikace, veškerou její dokumentaci a aplikace je nasazena do produkčního systému.

## 4.3 Úvod do vývoje v jazyce ABAP

Vývoj v jazyce ABAP se od vývoje v ostatních jazycích, jako je C nebo Java, poněkud liší. Pro lepší pochopení a zvládnutí základů jazyka ABAP je třeba porozumět aplikačnímu serveru a procesům, které na něm běží.

### 4.3.1 Vývoj na straně serveru

V jazyce ABAP se od počátku programuje na straně serveru. Programátor nevyvíjí aplikace na lokálním počítači, ani neuchovává zdrojové kódy v lokálním verzovacím systému. Pro vývoj aplikačních programů je potřeba přístup do SAP NetWeaver Application Ser-

ver<sup>21</sup> ABAP a vývojářské oprávnění. Instalace zkušební verze tohoto serveru a první přihlášení s vývojářským oprávněním je popsána v příloze A. Aplikační programy<sup>22</sup> se píšou v nástrojích ABAP Workbench, které jsou rovněž součástí serveru. Vývoj v jazyce ABAP je spojen se serverem nejen z pohledu vývojového prostředí a nástrojů. Jakmile se aktivuje zdrojový kód, vygeneruje se tzv. bajtkód, nezávislý na platformě a je přeložen v běhovém prostředí pro vykonání programu. Tento vygenerovaný kód programu je pak v aplikačním serveru (SAP NW AS ABAP) spuštěn. Další specifickou záležitostí je, že zdrojové kódy nejsou uchovávány v souborech. Jakmile se program uloží např. v nástroji ABAP editor, jsou zdrojové kódy a jejich verze uloženy do databáze, kde jsou pak přístupné pod názvem programu.

### 4.3.2 Třívrstvá architektura

SAP NetWeaver Application Server ABAP se skládá ze tří vrstev – *aplikační, prezentační a databázová*.

**Aplikační vrstva** leží mezi prezentační a databázovou vrstvou. Skládá se z jednoho nebo více aplikačních serverů a jednoho serveru, který odpovídá za komunikaci mezi vrstvami.

**Databázová vrstva** se skládá z centrální databáze (databázového systému) a vlastní databáze. Neobsahuje pouze uživatelská data, ale také veškerý programový kód SAP NetWeaver Application serveru a aplikačních programů, všechna administrativní data a nastavení parametrů (konfigurace).

**Prezentační vrstva** představuje rozhraní při komunikaci s uživateli. Tato vrstva přijímá uživatelské akce, což jsou kliknutí myši a vstupy z klávesnice, a posílá je do aplikační vrstvy. Potom přijímá data z aplikační vrstvy a zobrazuje je uživatelům.

Význam vrstev spočívá zejména ve škálovatelnosti. Rozdělení vrstev je čistě logické a neurčuje, do kolika fyzických strojů mají být uspořádány<sup>23</sup>. V praxi bývá však databázová vrstva na jednom stroji, který obsahuje uživatelská, programová a konfigurační data. Aplikační vrstva je rozprostřena na několik počítačů podle požadavků na výkon.

---

<sup>21</sup> dále jen SAP NW AS ABAP

<sup>22</sup> tak se říká aplikacím, které je možné na serveru vytvořit.

<sup>23</sup> Všechny vrstvy mohou být klidně na jednom počítači. To je však případ pouze mini-instalace systému SAP, sloužící pro výukové účely, a která je popsána v příloze A

### 4.3.3 Pracovní procesy

Aplikační programy běží v tzv. pracovních procesech (*Work processes*) v aplikační vrstvě, kde se zpracovávají. Pracovní procesy hrají roli **uživatelů v databázovém systému**, tj, každý pracovní proces má přiřazeno jedno spojení s databází. Pro vývojáře to znamená, že má připojení k databázi vždy po ruce. Nemusí se starat o vytváření, obsluhu a ukončování spojení – to vše je vyřešeno na straně serveru.

Při spuštění systému SAP je vytvořeno předem definované množství pracovních procesů. Vzhledem k tomu, že architektura systému SAP nemá technická omezení na počet uživatelů, kteří se mohou do systému přihlásit, nemůže být uživateli pevně přiřazen jeden pracovní proces. O rozdělování požadavků uživatelů na volné pracovní procesy se stará speciální podsystém *dispečer*. Dispečer optimalizuje požadavky uživatelů tak, aby pokud možno jeden pracovní proces zpracovával všechny po sobě jdoucí požadavky. Pokud to není možné, jednotlivé požadavky přiřadí různým pracovním procesům.

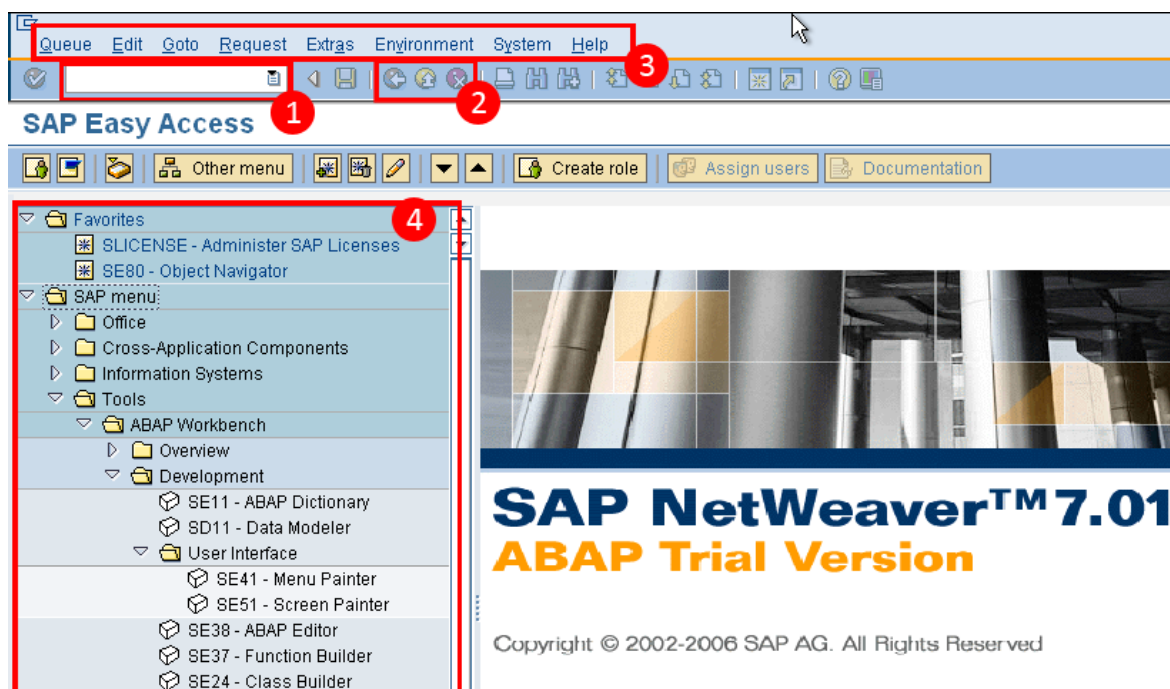
Pracovní procesy mohou být různého **typu**. Protože mají všechny pracovní procesy stejnou strukturu (viz dále kap. **Chyba! Nenalezen zdroj odkazů.**), tak jejich typ neurčuje, jaké jsou technické vlastnosti serveru, ale jaké úlohy (požadavky) se na serveru vykonávají. Dispečer založí pracovní proces a přiřadí mu typ podle úlohy, kterou má vykonat. Tím je možné distribuovat typy pracovních procesů na jednotlivé aplikační servery a optimalizovat využití jejich zdrojů. Každý server se může specializovat třeba na určitý typ pracovního procesu, ale na jednom serveru se mohou vyskytovat i všechny jejich typy:

- **Pracovní proces dialogů** (*Dialog Work Process*) – zpracovává požadavky od uživatelů a vykonává dynamické programy (viz dále), které se skládají z posloupnosti obrazovek.
- **Pracovní proces aktualizace** (*Update Work Process*) – zajišťuje vykonání příkazů UPDATE v databázi tak, aby byla zachována integrita. Souvisí s transakcemi na úrovni databáze, které se v systému SAP označují jako LUW (*Logical Unit Of Work*). Ty obsahují příkazy potvrzení a odvolání (COMMIT a ROLLBACK), aby bylo možné se vrátit k původnímu stavu v případě havárie.
- **Pracovní proces probíhající v pozadí** (*Background Work process*) – zpracovává programy, které mohou být vykonány bez interakce s uživatelem.
- **Řadící pracovní proces** (*Enqueue Work Process*) – spravuje tabulku zámků ve sdílené oblasti v paměti. Tabulka zámků obsahuje logické zámkové databáze v SAP NW AS ABAP. Zámky jsou zárukou toho, že není možné např. upravovat jeden záznam dvěma uživateli zároveň.
- **Tiskový pracovní proces** (*Spool Work Process*) – Připravuje data na tisk a odesílá je do tiskáren.

Obsahuje-li jeden server mnoho dialogových pracovních procesů a druhý mnoho procesů běžících v pozadí, nabízí se možnost nechat běžet první server přes den a druhý v noci.

## 4.4 Práce v SAP NW AS ABAP

Pro další výklad je nyní nezbytné se seznámit se základy ovládání uživatelského rozhraní aplikačního serveru, v jehož nástrojích se realizuje zákaznický vývoj.



**Obrázek 12:**

*Hlavní obrazovka SAP NW AS ABAP*

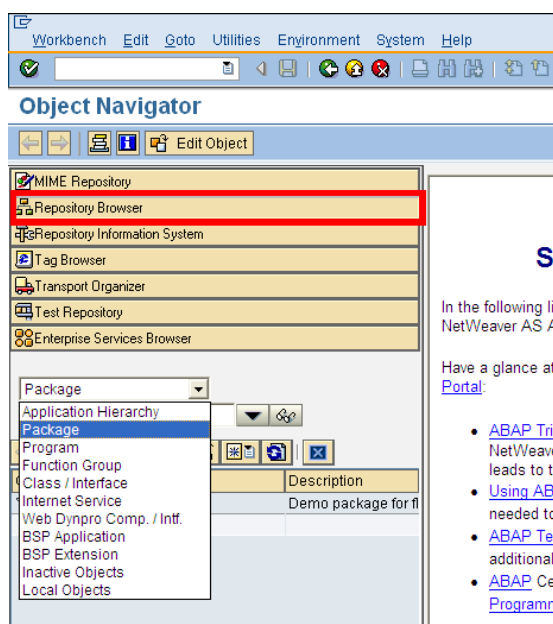
Úvodní obrazovku SAP NW AS ABAP ukazuje Obrázek 12. Nyní popíši hlavní oblasti této obrazovky:

1. Číslo 1 označuje pole, kam se zadávají **kódy transakcí**. Jakýkoliv program v systému SAP může mít přiřazen transakční kód, který tento program rychle spustí, aniž by uživatel musel procházet stromovou nabídku SAP. Transakce **SE38** např. spouští nástroj Editor ABAP.
2. Pod číslem 2 jsou na obrázku označeny **navigační ikony**. Vlevo je ikona Zpět, která zobrazí předchozí obrazovku. Uprostřed je ikona Konec, která přeruší právě prováděnou transakci a přesune uživatele na obrazovku zadání údajů. Vpravo je ikona Zrušit, která zruší celou transakci.
3. Číslo 3 označuje **kontextovou nabídku**, která má však v každé spuštěné aplikaci jiný význam. Jinde, než na úvodní obrazovce SAP NW AS ABAP, také nelze přímo zadávat kód transakcí. Mimo hlavní obrazovku je proto třeba psát kódy transakcí s lomítkem. Pokud je uživatel např. v nástroji ABAP editor a chce přejít do nástroje Data Dictionary, musí zadat název transakce jako /SE11.
4. Číslo 4 označuje stromovou nabídku SAP, který představuje alternativu při spuštění transakcí.

### 4.4.1 Správa objektů

Jednou z nejčastějších transakcí, které vývojář při práci v SAP NW AS ABAP, je transakce **SE80** – správa objektů (*Object Navigator*). Slovo „objekt“ může mít v terminologii SAP různé významy. Může být označením pro třídu nebo její instanci při objektovém programování, mnohem častěji se však používá jako označení pro jakékoliv programy a jejich součásti, které jsou vytvářeny během vývoje.

Správa objektů je místo, kde je možné přistupovat ke všem vývojovým objektům z celého systému. Mezi tyto vývojové objekty patří programy, moduly, třídy, rozhraní, ale i položky datového slovníku (viz dále), jako např. databázové tabulky, datové prvky apod. Objekty jsou součástí úložiště (*Repository*) a je možné je procházet pomocí průzkumníka (*Repository Browser*). Ten poskytuje přehled všech důležitých objektů a zobrazuje je dle různých kritérií, jak ukazuje Obrázek 13.



**Obrázek 13:**

*Správa objektů v nástroji ABAP Workbench*

Všechny objekty, které spolu logicky souvisí, je třeba nějak svázat, aby mohly být najednou přenášeny mezi jednotlivými systémy (viz další sekce) a byly součástí jakési softwarové logistiky. Proto je každý objekt součástí určitého **balíčku**<sup>24</sup> (*Package*). Výjimku tvoří balíček s názvem \$TMP. Tento balíček je standardně nadefinován v prostředí SAP NW AS ABAP a nikdy se nepřenáší mezi systémy. Je spojen pouze s uživatelem, který ho vytvořil, což znamená, že každý uživatel má svůj vlastní lokální balíček. Objekty v něm uložené se nazývají lokální objekty (*Local Objects*) a slouží zejména pro testovací a výukové účely.

<sup>24</sup> V české lokalizaci systému SAP se lze setkat také s označením „paket“, který je též použit v knize (Kühnhauser, 2012). V práci (Koubek, 2006) je naopak použito označení „vývojová třída“.

Při zákaznickém vývoji je také třeba dodržovat jmenné konvence pro vytváření nových objektů. Jednak proto, aby nedošlo ke konfliktům mezi již existujícími objekty, ale také aby bylo na první pohled jasné, které objekty byly vytvořeny vývojáři na straně zákazníka. Názvy objektů musí mít předponu Z\_ nebo Y\_, případně vlastní předponu, která je rezervována pro daného zákazníka.

## 4.5 Transportní systém

V rámci jednoho podniku může být vytvořeno několik systémů SAP, které obsahují klienty (viz kapitolu 2.3.4). Při přihlášení do systému, je třeba zadat trojmístné číslo klienta. Každý klient má **odlišné nastavení parametrů** a vlastní sadu kmenových dat, např. pro různé pobočky v rámci podniku. Přenos změn (např. úprava konfigurace nebo nová aplikace) z jednoho klienta do druhého se realizuje přes tzv. *transporty*, které zajišťuje **transportní systém**<sup>25</sup> (*Change and Transport System, CTS*).

### 4.5.1 Typy klientů a systémů SAP

Při vytváření klienta je třeba mu přiřadit aktivity, které v nich lze provádět, a tím získává klient **určitou roli**. Pro každou roli je vždy **přiřazen právě jeden klient**. Existují tři centrální role klientů, které se povinně vyskytují v každém systému:

- *Customizing and Development Client* (CUST) – zde je možné nastavovat parametry systému SAP, aby odpovídal podnikovým požadavkům, vyvíjet aplikace v ABAP Development Workbench a jinak upravovat systém.
- *Quality Assurance Client* (QTST) – v tomto klientu se testují nastavené parametry a vyvinuté aplikace ještě před tím, než jsou nasazeny do ostrého (produkčního) provozu.
- *Production Client* (PROD) – klient, který se používá pro vlastní užití systému SAP. Žádné úpravy parametrů nebo vývoj aplikací zde nejsou povoleny.

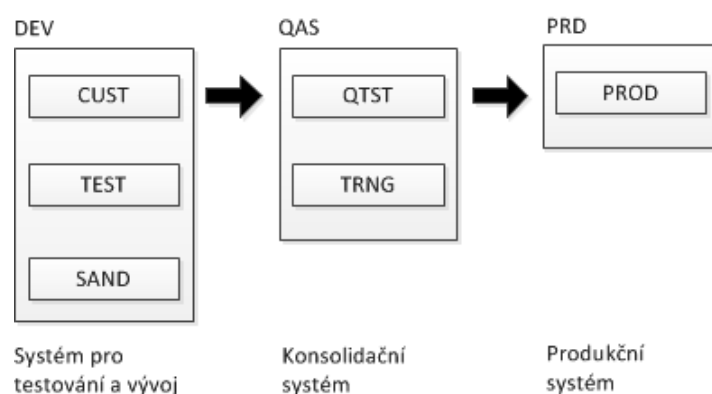
Kromě těchto centrálních klientů lze vytvářet **další klienty** podle potřeb podniku. V praxi se vyskytují klienti, jako je *Development Test Client* (TEST), který slouží k tvorbě testovacích aplikací. Dále se používá *Prototype and Sandbox Client* (SAND). Zde se zkoušejí úpravy parametrů, když si vývojář není jist, zda by měly být zahrnuty do hlavní funkcionality. *Training Client* (TRNG) je pak klient, kde si koncoví uživatelé zkoušejí novou funkcionality, která je pak přenesena do produkčního klienta. Je třeba mít ale na paměti, že

---

<sup>25</sup> V překladu knihy (Kühnhauser 2012) je transportní systém označen jako „*opravný a přenosový systém*“. Pro zjednodušení používám v této práci kratší název „*transportní systém*“, nebo zkratku CTS.

každý další klient **spotřebovává systémové zdroje** (paměť a prostor v databázi). Kromě toho se zvyšují nároky na administraci a roste složitost při aplikování změn z jednoho klienta do druhého pomocí CTS.

Když je rozhodnuto, které klienty bude celkový systém používat, je třeba vyřešit, jak budou **rozprostřeny** do jednotlivých **systémů SAP**, což se označuje jako tzv. *system landscape*. Všechny klienty lze použít i v jednom systému, což ale není doporučeno kvůli tomu, že jeden systém používá mnoho parametrů, jejichž změny se projevují napříč všemi klienty (*cross-client settings*), to značně omezuje flexibilitu systému a běh produkčního klienta současně s vývojem aplikací. Nejčastěji se lze setkat s klienty, distribuovanými do **3 systémů**, kde každý z centrálních klientů (CUST, QTST, PROD) má svůj vlastní systém (viz Obrázek 14).



**Obrázek 14:**

*Rozložení centrálních klientů do 3 systémů SAP. Zdroj: upraveno dle help.sap.com<sup>26</sup>*

Zde je charakteristika těchto tří systémů:

- **Systém pro testování a vývoj** (*development system, DEV*) ukládá zkušební data. Jak vyplývá z jednotlivých klientů (popsaných výše), používá se pro systémová nastavení, pro programování a testování. Jsou zde uložena nastavení pro daného zákazníka při procesu kastomizace. Ve své povaze jsou tyto systémy otevřené a nechráněné, obsahují spousty uživatelů s rozsáhlými oprávněními, kteří zde simulují a testují podnikové procesy.
- **Konsolidační systém** nebo též **systém zajišťující kvalitu** (*Quality Assurance System, QAS*) zajišťuje kvalitu vývoje a jsou zde kladeny vyšší požadavky na zabezpečení dat. Konsolidační systém se často používá pro akceptační testy, kde se ověřuje, zda nastavení systému odpovídá požadavkům podniku. Je-li vše v pořádku, transportují se všechny objekty do produkčního systému.



- **Produkční systém** je systém v ostrém provozu, kde pracují koncoví uživatelé a provádějí své transakce. Tento systém představuje pro podnik největší riziko, a tak je zde zajištěna nejvyšší ochrana. V produkčním systému se nikdy nevyvíjí aplikace ani nemění parametry

Na závěr je třeba dodat, že způsob rozložení klientů do systémů (*system landscape*) velmi záleží na **potřebách podniku**, jeho velikosti a předmětu podnikání. Uvedené rozložení tedy může být v praxi úplně jiné.

#### 4.5.2 Princip přenosu objektů mezi systémy

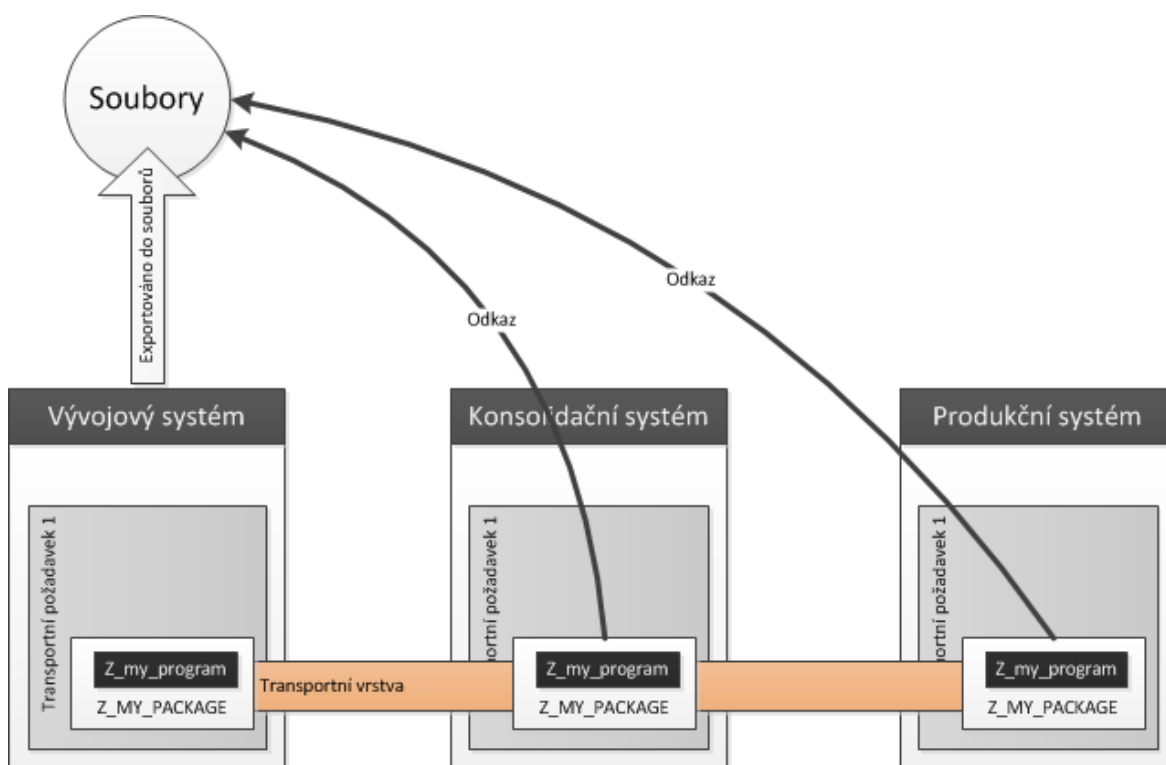
**Transportní systém** zajišťuje, že se správně promítnou změny provedené v klientu jednoho systému do klienta jiného systému. Změnou se rozumí jakákoliv změna – od upravení parametru systému až po vyvinutou aplikaci v ABAP Workbench. CTS zaznamenává všechny změny v tzv. *požadavcích* od vývojářů (*change requests*).

Jakmile vývojář vytvoří balíček, do kterého ukládá objekty, musí nastavit balíčku požadavek a transportní vrstvu (*Transport layer*). V analogii vlakové dopravy představuje požadavek vlakovou soupravu a transportní vrstva koleje, po kterých vlak jede. Přiřazení balíčku k požadavku znamená nastoupení na tento vlak. U každého balíčku je tedy specifikována transportní vrstva a označení požadavku. Každý vytvořený objekt musí být součástí nějakého balíčku.

Když vývojář dokončí svoji práci, provede uvolnění požadavku (*Request release*), doplní popis svého požadavku a program je transportován. Prakticky to znamená, že došlo k vytvoření fyzických souborů programu, které lze naimportovat do jakéhokoliv jiného systému. Programátor tedy nikdy nemá přímý vliv na cílový systém v tomto procesu. Odpovědná osoba s přístupem do cílového systému a jeho klienta pak rozhodne, zda program naimportuje nebo nikoliv. Transporty podléhají pravidlům definovaným v konfiguraci CTS. Jedním z pravidel může být to, že změny provedené v systému pro testování a vývoj, musí být nejprve transportovány do konsolidačního systému, a pak teprve do produkčního. Všechny změny jsou logovány, takže lze vždy dohledat, které požadavky byly importovány do kterého systému a jeho klienta, kterým uživatelem a zda došlo k nějakým chybám.

Po transportu programu z vývojového do konsolidačního systému není program duplikován, ale v novém systému se udržuje pouze odkaz na program původní. Transportovaný program je tedy uložen fyzicky v souboru pouze jednou. Když je nutné program změnit, vyvíjí se ten samý program opět ve vývojovém systému, ve stejném balíčku, ale je mu přiřazen nový požadavek. Ten je opět po dokončení práce uvolněn a v cílovém systému importován s tím, že předchozí požadavek je zachován. A právě na tomto principu funguje verzování aplikací. Kdykoliv je uvolněn požadavek, tak se zároveň uloží aktuální stav pro-

gramu a je možné se k němu kdykoliv vrátit. Princip transportu objektů je objasňuje Obrázek 15.



**Obrázek 15:**

*Schéma transportu objektů mezi systémy. Zdroj: upraveno dle help.sap.com*

## 4.6 Vývoj aplikačních programů

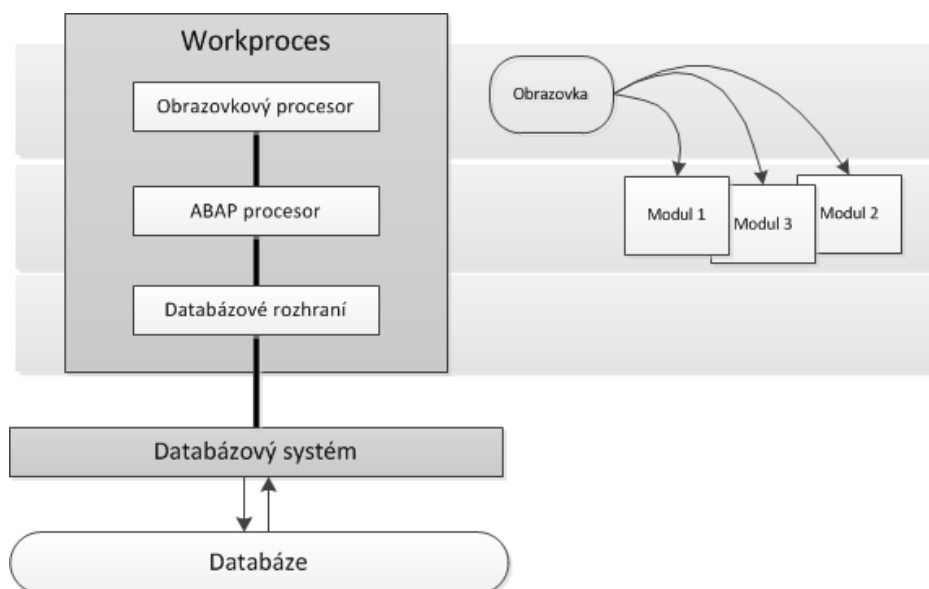
Jak už bylo zmíněno dříve, aplikacím vyvíjeným v jazyce ABAP se říká aplikační programy. Komponenty aplikačních programů, které umožňují komunikaci s uživatelem přes SAP GUI, jsou implementovány pomocí **obrazovek** (*screens*). Systém SAP rozlišuje 2 druhy aplikačních programů – *reporty* a *dynamické programy* (Kühnhauser, 2012).

**Reporty** jsou programy, které generují seznamy a rozhraní pro získávání dat. Důležitou vlastností reportu je to, že uživatel nijak neovlivní chování programu. Pouze nastaví počáteční kritéria a report mu poskytne data. Jsou tvořeny zpravidla jednou obrazovkou, které se říká **výběrová obrazovka** (*Selection screen*), kde uživatel nastavuje omezení pro výpis reportu.

**Dynamické programy** jsou tvořeny posloupností obrazovek a nelze předem určit, jaký bude průběh programu, protože ho určuje uživatel svým pohybem po obrazovkách (viz dále). Podle pojmenování druhu programu se obrazovce též v praxi říká **dynpro** (*Dynamic Program*). Toto označení používá např. (Kühnhauser 2012), lze se však setkat i s označením dialog.

Aplikační programy jsou zpracovány ve workprocesu, který obsahuje následující prvky:

- **Obrazkový procesor** (*screen processor*) – Všechny aplikační programy mají *logiku toku obrazovek* a *logiku zpracování*. Obrazkový procesor zodpovídá právě za **logiku toku obrazovek** a prostřednictvím dispečera komunikuje s workprocesy a SAP GUI. Na základě reakcí od uživatele jsou volány jednotlivé dialogové moduly aplikačního programu, které tvoří tok obrazovek (viz dále kap. 4.6.3 Dialogové programování).
- **Procesor ABAP** – zodpovídá za **logiku zpracování** aplikačního programu. Od obrazkového procesoru přijímá uživatelská data a zapisuje je do databáze přes databázové rozhraní a naopak. Dále zpracovává aritmetické a logické operace. Obrazkový procesor říká ABAP procesoru, který modul z logiky toku obrazovek má být zpracován. Komunikaci mezi procesory ukazuje Obrázek 16.
- **Databázové rozhraní** – přes toto rozhraní se přistupuje k databázi pomocí jednotné sady příkazů nezávisle na konkrétním databázovém systému.



**Obrázek 16:**

*Komunikace mezi obrazkovým procesorem a ABAP procesorem. Zdroj: upraveno dle help.sap.com<sup>27</sup>*

#### 4.6.1 Struktura programů v jazyce ABAP

Každý program začíná úvodním příkazem, který závisí na typu programu. Spustitelný program začíná úvodním příkazem REPORT <název programu>. To ale neznamená, že musí jít

27

[http://help.sap.com/saphelp\\_nw2004s/helpdata/en/fc/eb2e7d358411d1829f0000e829fbfe/content.htm](http://help.sap.com/saphelp_nw2004s/helpdata/en/fc/eb2e7d358411d1829f0000e829fbfe/content.htm)

výhradně o report – může to být i jiný druh aplikačního programu, např. dynamický program. Název příkazu „report“ je přežitek z dob, kdy se v jazyce ABAP tvořily pouze reporty.

Následují dvě pevně dané části programu – *deklarační část* a *bloky zpracování*.

**Deklarační část** definuje vše, co je možné globálně v programu využívat. Definují se zde použité databázové tabulky, datové typy, deklarace výběrových obrazovek, deklarace vnitřních tříd (pokud se využívá objektového přístupu). U rozsáhlých programů se tyto deklarace ukládají do tzv. top-include programů a jejich odkaz vložen na začátku programu.

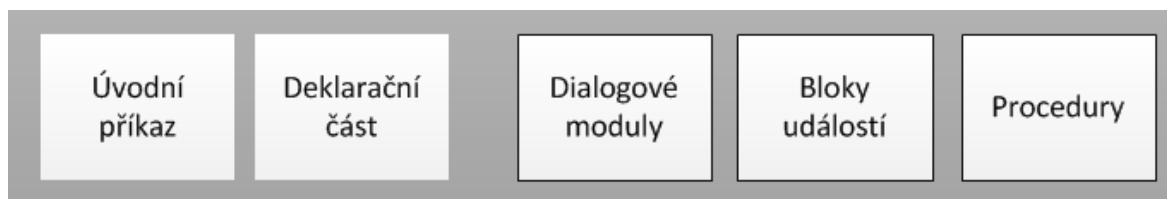
**Bloky zpracování** jsou modulární jednotky, které obrazovkový procesor (viz Obrázek 16:) volá podle daných pravidel, jak je určí vývojář. Sekvenčně je zpracován pouze kód uvnitř bloků zpracování. Mezi bloky zpracování patří:

- **Dialogové moduly** – bloky kódu volané při zpracování toku obrazovek. Jsou uzavřeny do párového příkazu MODULE a ENDMODULE.
- **Bloky událostí** – na začátku události je klíčové slovo pro událost. Příkladem události může být kliknutí myši nebo stisknutí tlačítka na obrazovce. Nastane-li nějaká událost, je tento blok zpracován.
- **Procedury** – opakovatelné části kódu obsahující rozhraní a lokální data. Patří sem:
  - **podrutiny**<sup>28</sup> (*Subroutines*) – podprogramy, které se volají v rámci jednoho programu. Jsou uzavřeny do párového příkazu FORM a ENDFORM. Volají se pak příkazem PERFORM. Slouží zejména pro zpřehlednění programu.
  - **funkční moduly** (*Function modules*) – funkce, které se volají (na rozdíl od podrutin) z programu externě. Nejsou tedy součástí programu, ale tzv. funkčních skupin (*Function groups*). Jsou uzavřeny do párového příkazu FUNCTION a ENDFUNCTION. Volají se pak příkazem CALL FUNCTION.
  - **metody** (*Methods*) – funkce, které jsou ale součástí tříd při objektovém programování (viz dále). Jsou uzavřeny do párového příkazu METHOD a ENDMETHOD. Volají se pak příkazem CALL METHOD.

Struktura logiky zpracování aplikačního programu znázorňuje Obrázek 17.

---

<sup>28</sup> toto označení používá např. (Koubek, 2006)

**Obrázek 17:**

*Struktura logiky zpracování aplikačního programu. Zdroj upraveno dle help.sap.com*

Všechny příkazy uvnitř programu (mimo deklarační části) jsou součástí nějakého **bloku zpracování**. Pokud jsou v programu nalezeny příkazy, u kterých není určen blok zpracování, jsou automaticky přiřazeny do **bloku** START-OF-SELECTION, který je zpracován jako první.

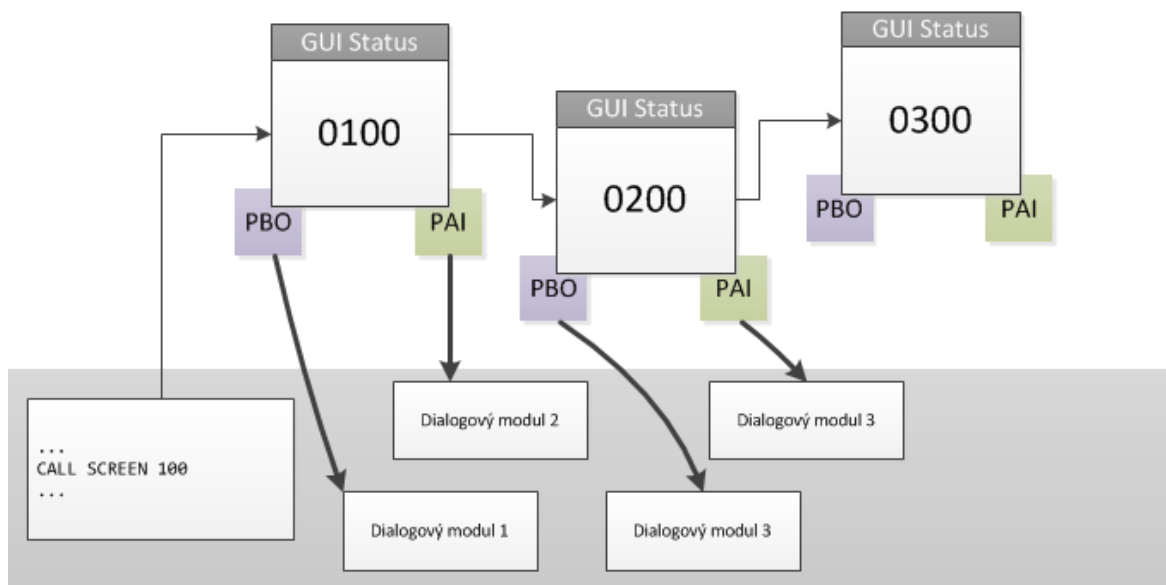
#### 4.6.2 Deklarace proměnných

V jazyce ABAP se deklarovaným proměnný říká **pole** (*Fields*) nebo **datové objekty**. Vedle toho však mohou existovat pole s neměnným obsahem, která se označují jako konstanty. Pole mohou být definovány pomocí vestavěného **datového typu** (např. celé číslo *i*) nebo pomocí vlastního **datového prvku** (viz dále). Polím je také třeba přiřadit nějaký název. Zde je příklad definice proměnné s názvem number, která je celočíselného typu.

```
DATA number TYPE i
```

#### 4.6.3 Dialogové programování

Dialogové programování je označení pro tvorbu **dynamických programů**, které jsou tvořeny posloupností obrazovek (dialogů, dynper). Každá obrazovka programu je jednoznačně určena číslem – např. 0100 a v programu se volá příkazem CALL SCREEN <číslo obrazovky>. Po zavolání obrazovky se nejprve vykoná akce PBO (*Process Before Output*), která obsahuje seznam modulů, které se mají vykonat před zobrazením obrazovky. Poté, co uživatel provede nějakou akci (např. stiskne tlačítko), vykoná se akce PAI (*Process After Input*), která obsahuje moduly, ve kterých lze reagovat na uživatelské vstupy. Tyto moduly pak mohou zavolat další obrazovku, ve které se opět nejprve vykoná PBO a tak dále. Schéma zpracování dynamického programu ukazuje Obrázek 18.

**Obrázek 18:**

*Schéma zpracování dynamického programu. Zdroj: upraveno dle help.sap.com<sup>29</sup>*

Obrazovka se vytváří pomocí nástroje *Screen Painter* (transakce **SE51**). Zde se nastavují také všechny části obrazovky:

- **Logika toku** – seznam modulů, které se mají zpracovat v akcích PBO a PAI
- **Seznam elementů** – tlačítka, vstupní pole a další GUI elementy „nakreslené“ v grafickém nástroji.
- **Atributy** – stejně jako každý objekt má i obrazovka nastaven balíček, popis, datum vytvoření, uživatele atd.
- **Layout** – zde se „kreslí“ rozvržení obrazovky pomocí grafického nástroje.
- **GUI status** – určuje, jaké ovládací ikony budou na obrazovce použity a jaké budou mít nastaven kód, aby bylo možné v modulu PAI reagovat na uživatelské vstupy. Je označen vlastním názvem a nastavuje se pomocí transakce **SE41**.
- **GUI titulek** – název obrazovky.

#### 4.6.4 Programování reportů

Reporty jsou dalším druhem aplikačních programů a jejich úkolem je načítat data z databáze a vypisovat je na obrazovku. Protože databázové tabulky mohou obsahovat i miliony záznamů, není vhodné (a někdy i reálné) vypsát všechny. Proto vývojář nadefinuje tzv. **výběrovou obrazovku** (*Selection screen*). Ta slouží k tomu, aby uživatel nastavil urči-

<sup>29</sup> [http://help.sap.com/saphelp\\_nw04/helpdata/en/9f/db9cdc35c111d1829f0000e829fbfe/content.htm](http://help.sap.com/saphelp_nw04/helpdata/en/9f/db9cdc35c111d1829f0000e829fbfe/content.htm)

tá omezení pro výpis, např. počet řádků, hodnoty pro některé sloupce a teprve poté program zpracuje příkaz `START-OF-SELECTION`. Zde může být zobrazení výběru naprogramováno různými způsoby. Může se např. použít obyčejný příkaz `WRITE`, který lze přirovnat např. k výpisu na konzoli v jazyce Java. Nejčastějším způsobem je však výpis řešen pomocí *ALV* (*ABAP List Viewer*), který dokáže zobrazovat data ve formě tabulky nebo stromu. Obsahuje navíc mnoho funkcí, které by jinak musel vývojář pracně programovat. Mezi hlavní funkce ALV patří:

- řazení záznamů podle sloupců,
- filtrování sloupců,
- hledání v záznamech,
- sčítání sloupců,
- zobrazení detailu řádku,
- a další.

Ukázka tvorby jednoduchého reportu je v příloze C.

## 4.7 Datový slovník

Datový slovník (ABAP Dictionary, transakce **SE11**) je nástroj, který umožňuje přistupovat k databázovému systému a je možné v něm pohodlně zakládat a měnit datové objekty slovníku. Mezi nejdůležitější typy objektů patří:

- **Tabulky** – databázové tabulky, které lze definovat i nezávisle na databázovém systému. Fyzicky jsou vytvořeny až po aktivaci tabulky. Každá tabulka obsahuje pole (Fields), která jsou určena datovým prvkem (viz dále). Existují 2 druhy tabulek:
  - klientově závislé – obsahují pole `MANDT`, které určuje klienta. Stejně tabulky pak mohou být použity v různých klientech (např. pobočkách firmy), ale s různými daty.
  - klientově nezávislé (cross-client) – v každém klientovi mají stejná data. Obsahují např. zdrojové kódy aplikačních programů, nastavení systému apod.
- **Pohledy** (Views) – logické pohledy na jednu nebo více databázových tabulek.
- **Zámky** (Lock objects) – používají se k synchronizaci přístupu k datům, kdy s daty pracuje více než jeden uživatel.

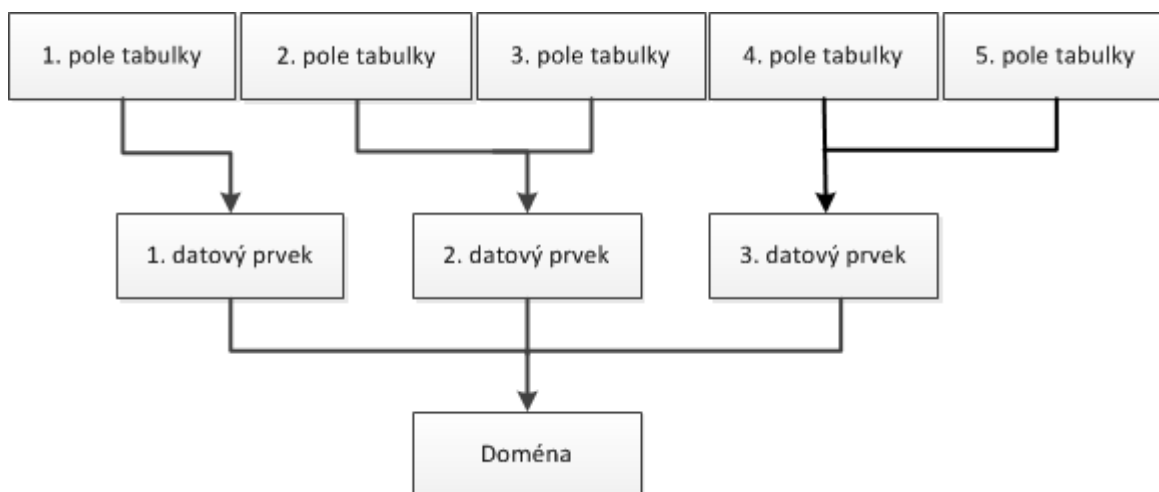
### 4.7.1 Datové prvky

Datový prvek je možné použít pro definici pole (sloupce) tabulky, ale i dalších struktur. Datový prvek může být 2 typů:

- referenčního typu – používá se k definici referenčních proměnných v objektovém rozšíření jazyka ABAP (*ABAP Objects*)
- primitivního typu – je definován vestavěným datovým typem a délkou. Např. datovým typem CHAR a délkou 10, které představuje řetězec o délce 10 znaků. Tyto hodnoty mohou být nadefinovány přímo, nebo se zkopírují z **domény**.

**Doména** určuje **omezení hodnot**, které může pole tabulky obsahovat. Např. pouze čísla od 1 do 10. V tomto případě má doména nastaveno, že je číselného typu a daného rozsahu. Doména může obsahovat definici přístupných hodnot také pomocí jejich výčtu. Jakmile je poli v tabulce přiřazen datový prvek, získává pole všechna nastavení a omezení definovaná v doméně.

Výhodou tohoto vztahu je znovupoužitelnost nadefinovaných domén a zejména zajištění konzistence při porovnávání jednotlivých polí mezi sebou. Vztah mezi datovými prvky, poli a doménami ukazuje Obrázek 19. Šipky znázorňují, čím je definováno pole tabulky. Více polí může mít přiřazen jeden datový prvek a více datových prvků může mít přiřazenu jednu doménu.



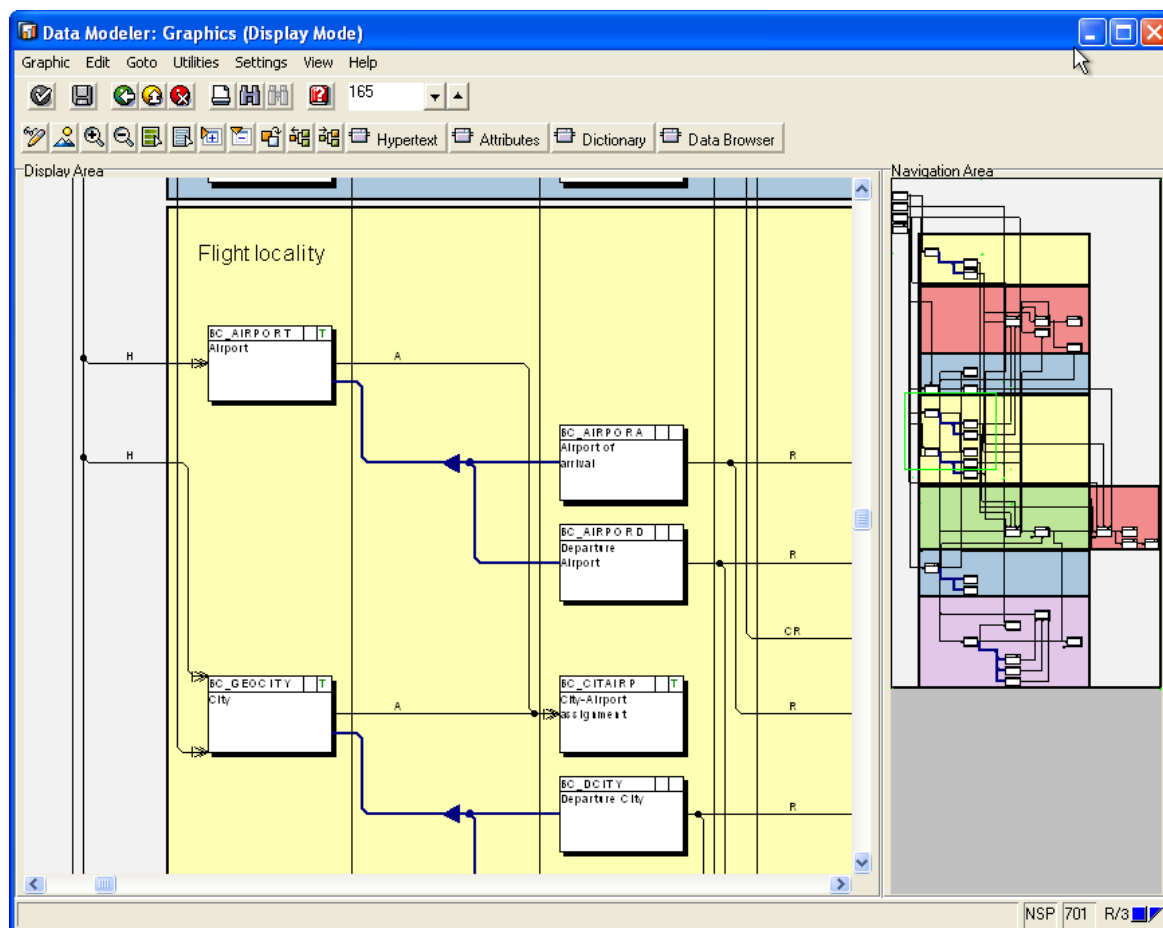
**Obrázek 19:**

*Vztah mezi poli tabulky, datovými prvky a doménami. Zdroj: autor.*

#### 4.7.2 Datové modelování

Užitečným nástrojem datového slovníku je také *Data modeler* (transakce **SD11**), který graficky znázorňuje vztahy mezi tabulkami (entitami). Ukázkový datový model (rezervace letů) je přístupný pod názvem BC\_TRAVEL. Grafický model se zobrazuje v menu *Utilities* → *Graphic* – viz Obrázek 20.





**Obrázek 20:**  
*Grafické znázornění datového modelu „rezervace letenek“*

## 5 Ukázková aplikace

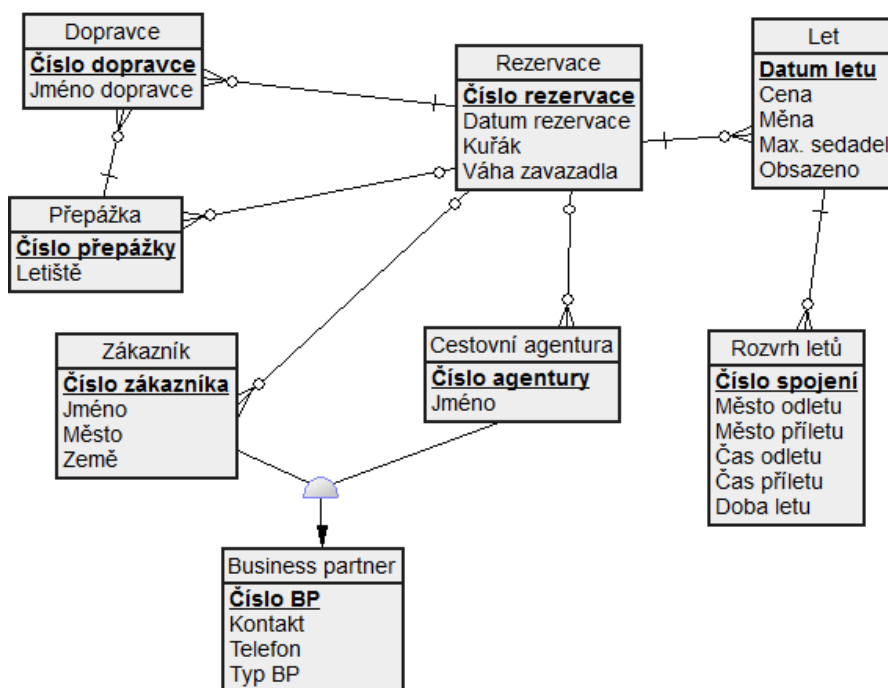
Tato kapitola demonstruje zákaznický vývoj v jazyce ABAP. Na jednoduchém zadání je prakticky ukázán postup tvorby aplikace a využití nástrojů z ABAP Workbench.

Cílem je vyvinout aplikaci v jazyce ABAP dle určité metodiky a vysvětlit její zdrojový kód.

### 5.1 Předpoklady a omezení ukázkové aplikace

Cílem této kapitoly je prakticky ukázat vývoj aplikace v systému SAP. Aby si vývoj aplikace mohl zkusit každý (nejen uživatel s přístupovými právy vývojáře do plné verze systému SAP) použil jsem zkušební verzi, též označovanou jako SAP-mini, která je zdarma na 30 dní a její instalace je popsána v příloze C. Použití této verze s sebou přináší určitá **omezení**, zejména z hlediska rozsahu transakcí, které lze v systému uskutečňovat. Ve zkušební verzi se totiž nenacházejí žádné obchodní transakce, protože je tvořena samotným NetWeaverem bez jakýchkoliv podnikových aplikací. Proto ukázka není napojena na standardní transakce, ale využívám datový model **rezervace letenek**, který je ve zkušební verzi systému SAP připraven pro **výukové účely**. Dalším omezením je existence pouze jednoho systému, a tak ukázka transportu objektů bude končit pouze uvolněním požadavku (*Transport Release*). Vytvořit druhý systém by znamenalo provést rozsáhlá nastavení transportu, což přesahuje rámec této práce.

**Cvičný datový model** se skládá z mnoha databázových tabulek, které uchovávají data související s leteckou dopravou. **Dopravce** představuje aerolinku, která se stará o přepravu osob. Rezervovat let může buď samostatná osoba, nebo cestovní agentura, která je významným business partnerem dopravce, protože pravidelně rezervuje mnoho letů. **Rezervace** může být uskutečněna na **přepážce** určitého letiště a určitého **dopravce**. Při rezervaci je přiřazen **let**, který je součástí **rozvrhu letů**. Obrázek 21 ukazuje zjednodušenou verze modelu.



Obrázek 21:

*Zjednodušený datový model cvičného příkladu. Zdroj: autor.*

## 5.2 Zadání ukázkové aplikace

Aplikace má vypisovat letecké spoje podle zadaných kritérií, kterým je dopravce a datum odletu. Uživatel má mít možnost řadit vzestupně a sestupně pro každý sloupec, dále vybírat jen určité sloupce k zobrazení a vyhledávat v celém výpisu.

Po kliknutí na řádek určitého spoje se zobrazí seznam všech letů, které jsou realizovány daným spojením.

## 5.3 Použití metodiky

Pro vývoj jakéhokoliv softwaru je dobré použít určitou **metodiku**. Metodiky představují souhrn postupů, pravidel a praktik, který poskytuje návod, jak při vývoji a následném provozu aplikace postupovat, kdo by se měl projektu účastnit a jaké výstupy by měly být vytvářeny. Společnost SAP používá svoji vlastní metodiku, která je však koncipována spíše pro vývoj **standardní funkcionality** systému SAP, a nikoliv pro zákaznický vývoj (Tuna, 2013).

Výběr metodiky pro zákaznický vývoj pak závisí na firmě, kde je systém SAP implementován. Použití metodiky je vhodné zejména z toho důvodu, že je-li vývoj pečlivě dokumentován, je později velmi snadné aplikace upravovat a udržovat. V této práci jsem pro ukázkový vývoj v systému SAP zvolil **metodiku MMSP**, kterou zde používám v rozsahu úměrnému velikosti aplikace.

Metodika MMSP vychází z metodiky OpenUP a je určena zejména pro menší softwarové projekty, které se realizují v rámci výuky na VŠE, ale i pro menší reálné projekty (délka vývoje do 6 měsíců). Z hlediska přístupu k vývoji se řadí, stejně jako OpenUP, mezi metodiky agilní (Rejnková, 2011). To znamená, že oproti tradičním (rigorózním) metodikám nepopisuje procesy, ale spíše principy a praktiky, není tolik zatížena dokumentací a jejím cílem je co nejrychlejší dodání fungujícího softwaru, který je dle zpětné vazby dále upravován (Buchalceková, 2005). Životní cyklus vývoje softwaru rozděluje metodika MMSP do čtyř fází, a to na fáze *Zahájení*, *Rozpracování*, *Konstrukce* a *Zavedení*.

Pro potřeby této práce a vzhledem k malému rozsahu aplikace, která slouží pouze pro demonstraci vývoje v systému SAP, je vždy provedena pouze jedna iterace v každé fázi. Při použití metodiky jsem vycházel ze struktury jejích dokumentů, ale nepoužil jsem všechny jejich části.

## 5.4 Fáze zahájení

Vizí projektu je vytvořit aplikaci v systému SAP, která vypíše seznam leteckých spojů (rozhvrh letů) a po kliknutí na konkrétní spoj zobrazí všechny lety, které probíhají v daném spoji.

### 5.4.1 Požadavky

Funkční požadavky, které má aplikace splňovat, jsou uvedeny v následující tabulce.

**Tabulka 5:** Funkční požadavky aplikace

ID požadavku	Název požadavku	Podrobný popis	Priorita	Odkaz na případ užití
F001	Vypsát letecké spoje	Vypsát všechny letecké spoje z databáze. Kritérium výpisu je název dopravce a datum odletu.	1	
F002	Vypsát realizované lety	Vypsát všechny konkrétní lety po kliknutí na záznam leteckého spoje. Zobrazit sloupce určující číslo spoje, datum letu, cenu, měnu, maximum sedadel, počet obsazených sedadel a celkový příjem z letu.	1	
F003	Zobrazit detail leteckého spoje	Po kliknutí na tlačítko se zobrazí detail označeného řádku ve výpisu spojů.	1	

ID požadavku	Název požadavku	Podrobný popis	Priorita	Odkaz na případ užití
F004	Možnost řadit tabulku	Záznamy ve výpisu leteckých spojů musí být možné seřadit vzestupně a sestupně kliknutím na záhlaví sloupce	1	
F005	Prohledávat tabulku	V záznamech leteckých spojů musí být možné vyhledávat	1	
F006	Vybírat sloupce	Pro přehlednost zobrazení záznamů leteckých spojů musí být možné vybrat jen určité sloupce, které budou zobrazeny	1	

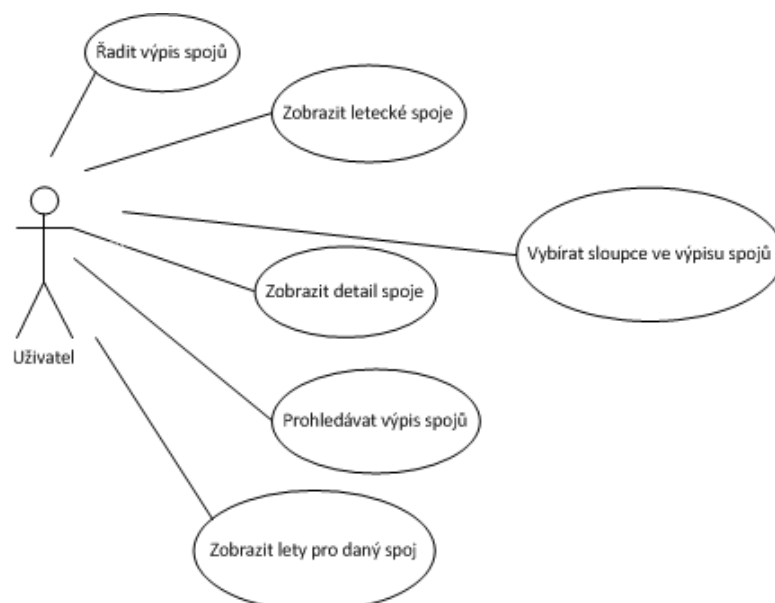
Nefunkční požadavky obsahuje následující tabulka.

**Tabulka 6:** Nefunkční požadavky aplikace

ID požadavku	Název požadavku	Podrobný popis	Priorita
NF001	Spouštět aplikaci zadáním kódu transakce	Pro urychlení práce by měla být aplikace přístupná prostřednictvím zadání kódu transakce	1

### 5.4.2 Případy užití

Na základě funkčních požadavků byly definovány případy užití, jak ukazuje následující obrázek.



**Obrázek 22:**

Diagram případů užití ukázkové aplikace. Zdroj: autor

**Tabulka 7:** Definice aktérů pro případy užití

Název aktéra	Popis aktéra
Uživatel	Aktér Uživatel je koncovým uživatelem systému SAP s oprávněním přístupu do produkčního systému. Uživatel používá aplikaci pro získání přehledu o rozvrhu letů, jaké lety jsou realizovány daným spojem, jaké je jejich obsazení apod.

**Tabulka 8:** Specifikace případů užití

Název	Id	Id požad.	Aktér	Cíl
Zobrazit letecké spoje	UC1	F001	Uživatel	Letecké spoje se zobrazí po zadání dopravce a času odletu. Pokud není zadán ani jeden z údajů, zobrazí se všechny letecké spoje.
Zobrazit detail spoje	UC2	F003	Uživatel	Zobrazit informace o spoji ve vyskakovacím okně po označení řádku a kliknutí na tlačítko „detail“.
Zobrazit lety pro daný spoj	UC3	F002	Uživatel	Po kliknutí na řádek spoje budou vypsány realizované lety v tomto spoji.
Řadit výpis spojů	UC4	F004	Uživatel	Po kliknutí na záhlaví sloupce ve výpisu spojů se výpis seřadí dle daného sloupce.
Prohledávat výpis spojů	UC5	F005	Uživatel	Po kliknutí na tlačítko „hledání“ se zobrazí vyskakovací okno, které bude obsahovat pole pro hledání. Po zadání posloupnosti znaků se zobrazí jen ty řádky, které obsahují zadané znaky.
Vybírat sloupce ve výpisu spojů	UC6	F006	Uživatel	Po kliknutí na tlačítko „vybrat sloupce“ se zobrazí vyskakovací okno, které nabídne seznam sloupců. Po označení určitých sloupců se zobrazí jen tyto označené.

## 5.5 Fáze rozpracování

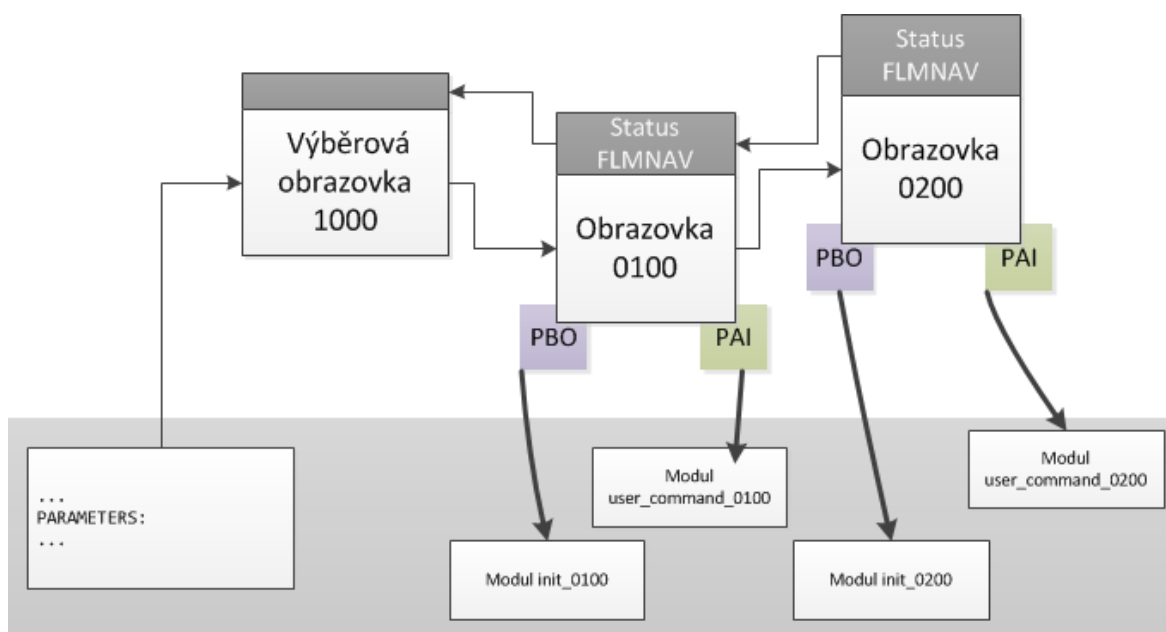
V této fázi je navrhována architektura aplikace – z jakých **modulů** se bude skládat a jaká bude posloupnost **obrazovek**.

Aplikace bude tvořena **třemi obrazovkami**. Jejich posloupnost bude zpracována obrazovkovým procesorem (viz Obrázek 16). Logika toku obrazovek a moduly, které budou v rámci nich volány, znázorňuje Obrázek 23.

- **Obrazovka** č. 1000 – výběrová obrazovka, kde se zadají kritéria pro výpis leteckých spojů.
- **Obrazovka** č. 0100 – bude obsahovat tabulku se seznamem leteckých spojů.

- **Obrazovka** č. 0200 – bude volána z obrazovky 0100 a zobrazí realizované lety.
- **Modul** `init_0100` – načte data z databáze a zobrazí je v seznamu na obrazovce. Bude zpracován v akci před zobrazením (PBO) obrazovky č. 0100.
- **Modul** `user_command_0100` – bude přijímat vstupy od uživatele, jako např. kliknutí na tlačítko zpět v navigaci okna. Bude zpracován v akci po uživatelském vstupu (PAI) na obrazovce č. 0100
- **Modul** `init_0200` – načte data z databáze a zobrazí je ve výpisu. Modul bude zpracován před zobrazením (PBO) obrazovky č. 0200.
- **Modul** `user_command_0200` – bude přijímat vstupy od uživatele, jako např. kliknutí na tlačítko zpět v navigaci okna. Bude zpracován v akci po uživatelském vstupu (PAI) na obrazovce č. 0200

Obrazovky 0100 a 0200 obsahují status s názvem FLMNAV, který je tvořen tlačítky zpět, konec a zrušit. Tlačítkům jsem v transakci SE41 přiřadil postupně kódy BACK, EXIT a CANCEL.



**Obrázek 23:**

*Logika toku obrazovek a moduly volané v akcích PBO a PAI*

Pro zobrazování interaktivních seznamů poskytuje SAP nástroj zvaný ALV (*ABAP List Viewer*). Použitím tohoto nástroje budou pokryty všechny požadavky ohledně práce s tabulkou. ALV je dostupný mimo jiné i prostřednictvím tříd z *ABAP Objects*, což je rozšíření jazyka ABAP o objektový přístup. Pro výpis leteckých spojů bude použita třída `cl_salv_table`. Je třeba také připravit třídu, která bude reagovat na kliknutí uživatele do tabulky a předávat číslo řádku jako parametr. K tomu bude vytvořena třída `cl_event_handler` a metoda `on_double_click`.

Na základě této architektury jsem vytvořil kód aplikace, který obsahoval definici třídy `cl_event_handler`, jak je vidět na následujícím výpisu.

```
CLASS cl_event_handler DEFINITION.  
  PUBLIC SECTION.  
    CLASS-METHODS: on_double_click FOR EVENT  
if_salv_events_actions_table~double_click OF cl_salv_events_table  
    IMPORTING  
      row  
      column.  
ENDCLASS.
```

A definici modulů, jak je vidět na tomto výpisu.

```
MODULE init_0100 OUTPUT.  
ENDMODULE.  
MODULE user_command_0100 INPUT.  
ENDMODULE.  
  
MODULE init_0200 OUTPUT.  
ENDMODULE.
```

V této fázi se také poprvé tvoří a spouští **jednotkové testy**, které jazyk ABAP podporuje<sup>30</sup>, ale vzhledem k malému rozsahu aplikace nejsou v této práci zahrnuty.

## 5.6 Fáze konstrukce

V této fázi jsem naimplementoval jednotlivé moduly tak, aby aplikace vyhovovala funkčním požadavkům a vycházela ze stanovené architektury.

Nejprve jsem vytvořil **obrazovky**, a do každé jsem v nástroji *Screen Painter* připravil elementy s názvem `VYPIS_SPOJU` a `VYPIS_LETU`, do kterého se načte ALV seznam. Do **logiky toku obrazovek** jsem nastavil, které moduly se mají spouštět v akcích PBO a PAI. Jako příklad uvádím logiku toku obrazovky č. 0100 na následujícím výpisu.

```
PROCESS BEFORE OUTPUT.  
  MODULE init_0100.  
  
PROCESS AFTER INPUT.  
  MODULE user_command_0100.
```

---

30

[http://help.sap.com/saphelp\\_nw73ehp1/helpdata/en/49/180615005338a1e10000000a421937/content.htm](http://help.sap.com/saphelp_nw73ehp1/helpdata/en/49/180615005338a1e10000000a421937/content.htm)



Poté jsem naimplementoval logiku všech modulů. Jak už bylo uvedeno v předchozí fázi, **modul init\_0100** vytváří objekt typu `cl_salv_table` a naplňuje ho daty z tabulky `SPFLI`. Předtím je ale třeba ještě vytvořit objekt typu `cl_gui_custom_container`, který zajistí prostor pro zobrazení seznamu. Jako parametr se mu musí předat název elementu z obrazovky, tedy `VYPIS_SPOJU`. Tím je hotovo zobrazení seznamu leteckých spojů v `ALV`.

**Modul user\_command\_0100** zpracovává vstupy od uživatele, konkrétně stisknutí navigačních tlačítek *BACK*, *EXIT* a *CANCEL*. Pro zpracování kliknutí do seznamu spojů je však třeba naimplementovat třídu `cl_event_handler` (viz dále).

**Modul init\_0200** zobrazuje seznam realizovaných letů dle zadaného klíče (sloupec `CONNID` v tabulce `SPFLI`). Po opuštění seznamu (stiskem tlačítka *BACK* nebo *CANCEL* v menu systému `SAP`) program přejde zpět k obrazovce č. 0100 (viz Obrázek 23).

Posledním krokem bylo získat parametr, který je číslem leteckého spoje (sloupec `CONNID`), na který uživatel klikl. Pomocí tohoto čísla je pak možné vypsát lety realizované tímto spojením. Za tímto účelem jsem naimplementoval metodu `on_double_click` třídy `cl_event_handler`, která se spustí ve chvíli, kdy uživatel klikne do seznamu leteckých spojů. Dle její definice (viz předchozí fáze) je metodě předán parametr pořadového čísla řádku a sloupce v seznamu. Pomocí pořadového čísla řádku jsem získal přístup ke všem polím daného řádku, tedy i k poli `CONNID`. Tuto hodnotu jsem uložil do paměti pomocí konstrukce `PARAMETER ID`, aby bylo možné k ní přistupovat v další obrazovce. Implementaci této metody ukazuje následující výpis:

```
CLASS cl_event_handler IMPLEMENTATION.  
  METHOD on_double_click.  
  
    DATA: schedule TYPE spfli.  
  
    READ TABLE itab INTO schedule INDEX row.  
    IF sy-subrc = 0.  
      connection = schedule-connid.  
    ELSE.  
      MESSAGE 'Nebyl nalezen zadny let' TYPE 'I'.  
    ENDIF.  
  
    SET PARAMETER ID 'CON' FIELD connection.  
    CALL SCREEN 0200.  
  
  ENDMETHOD.  
ENDCLASS.
```

Výběrová obrazovka je volána klíčovým slovem `PARAMETERS`, kde jsou nastavena pole, která určují omezení pro výpis leteckých spojů. Na základě předaných parametrů se pak načítají data do interní tabulky s názvem `itab`, která obsahuje letecké spoje jen pro zadaného dopravce a se zadaným nebo pozdějším datem odletu.

```

PARAMETERS: airline LIKE spfli-carriid,
            dep LIKE spfli-deptime.

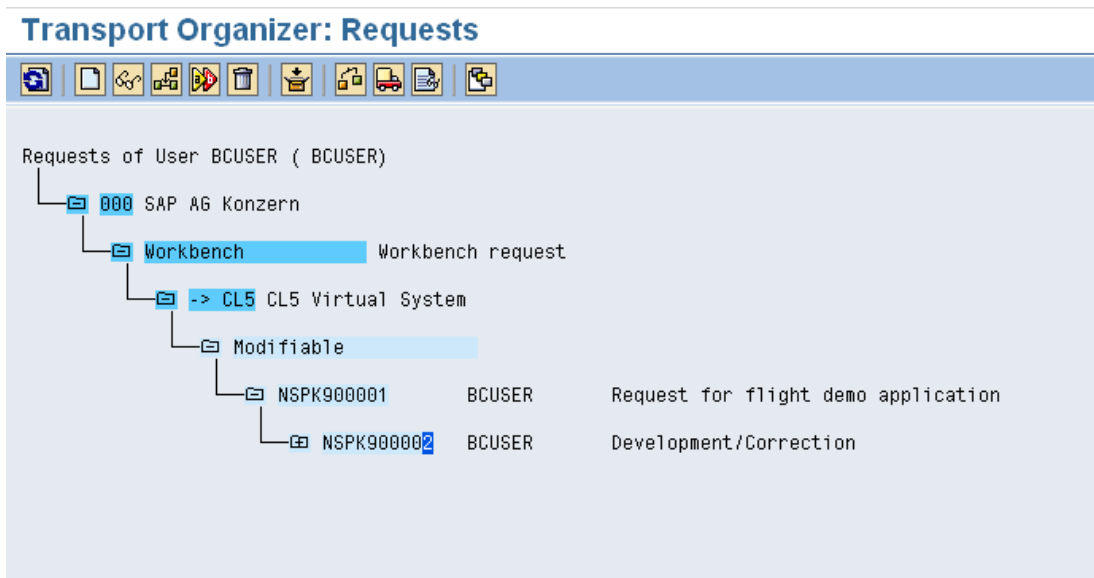
START-OF-SELECTION.

IF NOT airline IS INITIAL.
    SELECT * FROM spfli
    INTO TABLE itab WHERE carriid = airline AND deptime > dep.
ELSE.
    SELECT * FROM spfli
    INTO TABLE itab WHERE deptime > dep.
ENDIF.

CALL SCREEN 100.

```

Pro úplný výpis zdrojového kódu viz přílohu D. Tím jsem dokončil implementaci a vznikla beta verze programu. V tomto momentě se aplikace v praxi přenáší pomocí CTS do **testovacího systému** (konsolidačního systému), kde je testována uživateli. Vzhledem k tomu, že jsem od začátku vyvíjel aplikaci v balíčku Z\_DEMO\_FLIGHT, kterému je přiřazen požadavek a transportní vrstva, je celá aplikace připravena na transport do jakéhokoliv dalšího systému a jeho klienta. V nástroji *Transport Organizer* (transakce **SE09**) jsem odeslal požadavek ze zdrojového klienta 000. Na následujícím obrázku je požadavek uživatele BCUSER a seznam jeho vytvořených objektů ve vývojovém prostředí (*workbench request*). Požadavek jsem nazval **request for flight demo application** a obsahuje vytvořenou aplikaci.



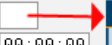
**Obrázek 24:**

*Aplikace jako součást požadavku, který je připraven k odeslání. Zdroj: autor.*

## 5.7 Fáze zavedení

Fáze zavedení je poslední fází životního cyklu vývoje dle metodiky MMSP, kdy je aplikace uvedena do provozu. V zákaznickém vývoji systému SAP to znamená přenést aplikaci do produkčního systému, kde je dostupná pro uživatele pod kódem transakce. Transport do produkčního systému probíhá obdobně jako při transportu z vývojového do testovacího systému. Výsledná aplikace je vidět na následujícím obrázku.

**Program Z\_FLMANAGER** **Obrazovka 1000**

Specify the airline  

Specify the departure time

**Restrictions**

ID	Airline
AA	American Airlines
AB	Air Berlin
AC	Air Canada
AF	Air France
AZ	Alitalia
BA	British Airways

**Obrazovka 0100**

Cli...	Air...	Flig...	Co...	Depart. city	De...	Co...	Arrival city	Ta...	Flight t...	Departure	Arrival	Dista
000	UA	941	DE	FRANKFURT	FRA	US	SAN FRANCISCO	SFO	11:36	14:30:00	17:06:00	5.
000	UA	3504	US	SAN FRANCISCO	SFO	DE	FRANKFURT	FRA	10:30	15:00:00	10:30:00	5.
000	UA	3516	US	NEW YORK	JFK	DE	FRANKFURT	FRA	7:25	16:20:00	05:45:00	6.
000	UA	3517	DE	FRANKFURT	FRA	US	NEW YORK	JFK	8:15	10:40:00	12:55:00	6.

**Obrazovka 0200**

Client	Air...	Flig...	Flight Date	Airfare	Currency	Plane Type	Capacity	Occupied	Booking total
000	UA	3517	25.02.2008	611,01	USD	747-400	385	365	276.219,65
000	UA	3517	24.03.2008	611,01	USD	747-400	385	363	274.160,54
000	UA	3517	21.04.2008	611,01	USD	747-400	385	363	272.070,91
000	UA	3517	19.05.2008	611,01	USD	747-400	385	373	277.881,59
000	UA	3517	16.06.2008	611,01	USD	747-400	385	372	281.003,87
000	UA	3517	28.06.2008	611,01	USD	747-400	385	368	274.960,97
000	UA	3517	14.07.2008	611,01	USD	747-400	385	371	279.073,08
000	UA	3517	28.07.2008	611,01	USD	747-400	385	367	275.871,38

**Obrázek 25:**

*Uživatelské rozhraní ukázkové aplikace. Zdroj: autor.*

## 6 Závěr

Cílem této diplomové práce bylo popsat a prakticky demonstrovat zákaznický vývoj v systému SAP.

Pro dosažení tohoto cíle bylo nutné nejprve uvést čtenáře do problematiky prostředí systému SAP. Proto jsem charakterizoval společnost SAP, její postavení na trhu, produktové portfolio a architekturu produktu SAP Business Suite. Společnost SAP je největším dodavatelem podnikových aplikací a jednou z největších softwarových firem na světě. V české republice jsou však podnikové aplikace implementovány převážně místními dodavateli. I v kategorii řešení pro velké podniky je nejrozšířenějším systémem produkt Helios. Systém SAP All-in-one je v této kategorii na druhém místě před systémem Microsoft Dynamics. Produktové portfolio společnosti SAP je velmi široké. Obsahuje nejen podnikové aplikace pokrývající funkcionalitu ERP II, ale také databázové systémy, datové sklady, cloudové a mobilní platformy, analytické aplikace a další.

Protože se práce zabývá programovými úpravami systému SAP, bylo dalším dílčím cílem charakterizovat technologický základ SAP NetWeaver. SAP NetWeaver lze chápat jednak jako základ, na kterém jsou postaveny jednotlivé aplikace, a také jako integrační platformu, která umožňuje propojovat systém SAP s dalšími systémy, rozšiřovat systém do prostředí internetu a skládat dohromady aplikace dle principů architektury orientované na služby. SAP NetWeaver obsahuje mnoho komponent, které jsou tvořeny aplikacemi a nástroji a zajišťují pružnou reakci na změny požadavků na funkcionalitu systému SAP. Popis každé komponenty zvlášť by nevystačil ani na celou knihu, a tak jsem se snažil poskytnout alespoň základní přehled komponent a jejich užití. Strategický význam SAP NetWeaver spočívá zejména ve snižování TCO, a to díky snadné integraci aplikací postavených na tomto základu a možnosti oddělit instalaci podnikových aplikací a technologického základu.

Dalším dílčím cílem bylo popsat již samotný zákaznický vývoj v jazyce ABAP a nezbytné nástroje, které se k tomu využívají. Vývoj v jazyce ABAP se realizuje v komponentě SAP NetWeaver Application Server, která tvoří základ celé platformy. Ta obsahuje vývojové prostředí ABAP Development Workbench, které obsahuje několik vývojových nástrojů. Principy vývoje a jednotlivé nástroje jsem popsal do takové míry detailu, aby byl čtenář schopen sám prakticky vše vyzkoušet a vytvářet aplikace, které se běžně při zákaznickém vývoji v jazyce ABAP programují. Přílohy této práce proto obsahují podrobné návody, jak s těmito nástroji pracovat včetně instalace bezplatné zkušební verze systému SAP (SAP mini).

Posledním cílem bylo vyvinout ukázkovou aplikaci v jazyce ABAP. K tomu jsem použil metodiku MMSP a celý vývoj zdokumentoval. Úplné zdrojové kódy aplikace jsem pak vložil do přílohy této práce. Mám-li shrnout vlastnosti jazyka ABAP, tak je určitě velmi „upovídaný“, což je dáno hlavně tím, že vznikl již v 80. letech minulého století. Nicméně i přes mírně zdlouhavé konstrukce mi tento jazyk v různých situacích připomínal ostatní

jazyky, na které jsem byl dosud zvyklý (Java, PHP). Křivku učení jazyka ABAP hodnotím jako velmi strmou, protože základy se dají naučit opravdu celkem rychle. Nicméně zvládnutí pokročilejších technik vyžaduje dlouhodobé praktické zkušenosti.

Na závěr konstatuji, že všech vytyčených cílů bylo splněno. Hlavním přínosem práce je, že utváří ucelený materiál, který poskytuje úvod do vývoje aplikací v systému SAP, včetně praktické ukázky programování v jazyce ABAP. Protože nejlepším způsobem, jak této problematice porozumět, je praktické vyzkoušení vývoje v tomto jazyce. Věřím, že tato práce je vhodným startem nejen pro vývojáře, kteří začínají s jazykem ABAP pracovat, ale také pro podnikové specialisty, kteří chtějí porozumět technologickému pohledu na systém SAP.

Práci bych si dovolil zakončit citátem, který vystihuje význam vlastního vyzkoušení vývoje v systému SAP oproti teoretickému studiu.

*„Zkušenost vybírá hrozně vysoké školné, ale učí jako nikdo jiný.“*

– Thomas Carlyle

# Terminologický slovník

Termín	Zkratka	Význam [zdroj]
Accelerated SAP	ASAP	Metodika, která je standardem pro popis časového plánu implementace systému SAP z hlediska řízení projektu. Pomáhá podnikům vysvětlit, co přinese změna jejich procesů, a to jak z podnikového, tak i z IT pohledu (Anderson 2012, s. 250)
Advanced Business Application Programming	ABAP	Původně Allgemeiner Berichts-Aufbereitungs-Prozessor – programovací jazyk, vyvinutý společností SAP, který slouží k vývoji podnikových aplikací (Kühnhauser 2012, s. 16)
Cloud computing		Použití výpočetních zdrojů (hardware a software), které jsou poskytovány jako služba přes internet (Anderson 2012, s. 315)
Custom development		Zákaznický vývoj, který je realizován při implementaci systému SAP v případě, kdy je třeba upravit funkcionalitu systému pomocí vývojových nástrojů, propojit systém s jinými aplikacemi nebo poskytnout jeho část pro architekturu SOA (SAP, 2013b).
Customer Relationship Management	CRM	Soubor aplikací, podnikových procesů a personálních zdrojů, které zajišťují řízení vztahů se zákazníky firmy, a to v oblastech podpory obchodních činností, zejména prodeje, marketingu a zákaznických služeb (Gála et al. 2009, s. 210)
Enterprise Resource Planning	ERP	Celopodnikový software, který pomocí informačních a komunikačních technologií pomáhá integrovat a automatizovat procesy v oblasti financí a logistiky (Basl, 2008 s. 66).
Change And Transport System	CTS	Změnový a transportní systém, mechanismus, který zajišťuje přenos změn (např. úprava konfigurace nebo nová aplikace) z jednoho systému SAP do druhého (např. z vývojového systému do produkčního) (Koubek 2006)
Metodika pro malé softwarové projekty	MMSP	Metodika vývoje softwaru, která byla vytvořena lokalizací a úpravou metodiky OpenUP především pro projekty probíhající v rámci výuky softwarového inženýrství na VŠE v Praze (Rejnková 2011)
Product Lifecycle Management	PLM	Podpora řízení životního cyklu výrobku zahrnující návrh, konstrukci, testování, dodávku, instalaci a údržbu (Gála et al. 2009, s. 179)

SAP Business All-In-One		Řešení od společnosti SAP pro podniky se 100 až 2500 zaměstnanci. Jedná se o úplné podnikové řešení, které vychází ze SAP ERP a obsahuje CRM řešení. Podporuje využití několika lokalit a různých typů poboček či dceřiných společností (Anderson 2012, s. 85)
SAP Business ByDesign		Řešení od společnosti SAP vhodné pro střední podniky od 100 do 500 zaměstnanců, dodávané formou SaaS (Anderson 2012, s. 83)
SAP Business One		Řešení od společnosti SAP, které nahrazuje několik izolovaných a různorodých aplikací jediným softwarem, který integruje CRM, výrobu a finanční účetnictví. Vhodné pro malé podniky do 100 zaměstnanců (Anderson 2012, s. 81)
SAP Business Suite		Nabídka softwarového řešení od společnosti SAP pro některé z největších a nejznámějších společností na světě. Obsahuje několik komponent, jejichž jádro tvoří SAP ERP (Anderson 2012, s. 75)
SAP Enterprise Resource Planning	SAP ERP	Následovník systému SAP R/3. Komponenta rodiny produktů SAP Business suite, která řeší každodenní správu financí a zdrojů velkých podniků (Anderson 2012, s. 33).
SAP NetWeaver Application Server	SAP NW AS	Komponenta platformy SAP NetWeaver. Základ, na kterém stojí celé řešení SAP Business Suite a také ostatní komponenty této platformy. Tato komponenta je úzce spjata se svým vývojovým prostředím, protože umožňuje programovat zákaznické aplikace a je zároveň běhovým prostředím, ve kterém jsou aplikace spouštěny (Woods, Word 2004a)
SAP NetWeaver	SAP NW	Technologický základ většiny dnešních produktů firmy SAP a umožňuje jejich provoz a rozvoj. Tento základ lze chápat i jako integrační platformu (Anderson 2012, s. 91)
SAP NetWeaver Master Guide		Příručka, která pomáhá systémovým administrátorům a technickým konzultantům s implementací systémů založených na platformě SAP NetWeaver (Anderson 2012, s. 100)
Service Oriented Architecture	SOA	Politiky, praktiky a rámce, které umožňují, aby funkcionality aplikací byla poskytována a spotřebována jako množina služeb, a to v takové úrovni granularity (rozsahu funkcionality), kterou potřebuje příjemce služby (Gála et al. 2009)
Supply Chain Management	SCM	Činnost spočívající v integraci organizačních jednotek, které tvoří dodavatelský řetězec v koordi-

		nací materiálových, informačních a finančních toků (Gála et al. 2009, s. 200)
Sytems Applications and Products in data processing	SAP	Původně Systeme, Anwendungen, Produkte in der Datenverarbeitung – společnost, která je největším dodavatelem podnikových aplikací a jednou z největších softwarových firem na světě. Svými produkty pomáhá podnikům všech velikostí získat náskok před konkurencí efektivním využitím informačních technologií (SAP 2013e).



## Seznam literatury

ANDERSON, George. 2009. *SAP Implementation Unleashed: A Business and Technical Roadmap to Deploying SAP*. 1. S.l.: Sams Publishing. ISBN 978-0672330049.

ANDERSON, George W. 2012. *Naučte se SAP za 24 hodin*. Přel. Milan DANĚK. Brno: Computer Press. ISBN 978-80-251-3685-0.

BASL, Josef. 2008. *Podnikové informační systémy: podnik v informační společnosti*. 2., výrazně přeprac. a rozš. vyd. Praha: Grada. Management v informační společnosti. ISBN 978-80-247-2279-5.

BERTINO, Elisa a Kenji TAKAHASHI. 2010. *Identity Management: Concepts, Technologies, and Systems*. S.l.: Artech House. ISBN 978-1-60807-039-8.

BRANDT, Wener. 2013. *Fourth Quarter and Full-Year 2012* [online]. 23. leden 2013. S.l.: SAP. Dostupné z: <http://www.sap.com/corporate-en/investors/pdf/SAP-2012-Q4-Presentation.pdf>

BUCHALCEVOVÁ, Alena. 2005. *Metodiky vývoje a údržby informačních systémů*. 1. vyd. Praha: Grada. Management v informační společnosti. ISBN 80-247-1075-7.

CLARK, William, Ian FINLEY a Song CHUANG. 2012. Magic Quadrant for Mobile Application Development Platforms. *Gartner* [online]. [vid. 16. březen 2013]. Dostupné z: <http://www.gartner.com/technology/reprints.do?id=1-1AAJYX9&ct=120427&st=sb>

ČSÚ. 2012. Využívání informačních a komunikačních technologií v podnikatelském sektoru. ČSÚ [online]. [vid. 18. březen 2013]. Dostupné z: <http://www.czso.cz/csu/2012edicniplan.nsf/p/9702-12>

FARBER, Dan. 2007. MySAP fades into history. *ZDNet* [online]. [vid. 4. březen 2013]. Dostupné z: <http://www.zdnet.com/blog/btl/mysap-fades-into-history/4901>

FERGUSON, Renee Boucher. 2003. SAP Drills for Developers ; Company extends reach of NetWeaver environment. *eWeek*. roč. 20, č. 36, s. 12 ISSN 15306283.

GÁLA, Libor, Jan POUR a Zuzana ŠEDIVÁ. 2009. *Podniková informatika*. 2. přeprac. a aktualiz. vyd. Praha: Grada. Expert. ISBN 978-80-247-2615-1.

GÁLA, Libor, Jan POUR a Prokop TOMAN. 2006. *Podniková informatika*. Praha: Grada Publishing a.s. ISBN 80-247-1278-4.

HELLAND, Tanner. 2012. Where does Microsoft make money? *Tanner Helland (dot) Com* [online]. [vid. 29. březen 2013]. Dostupné z: <http://www.tannerhelland.com/4273/microsoft-money-updated-2012/>

HESTERMANN, Christian, Chris PANG a Nigel MONTGOMERY. 2012. Magic Quadrant for Single-Instance ERP for Product-Centric Midmarket Companies. *Gartner* [online]. [vid. 12. březen 2013]. Dostupné z: <http://www.gartner.com/technology/reprints.do?id=1-1B4MV8J&ct=120628&st=sg>

CHALOUPKA, Vladimír. 2009. *Možnosti vývoje specializovaných aplikací v MS Dynamics AX*. S.l. Diplomová práce. Vysoká škola ekonomická v Praze.

CHICKOWSKI, Ericka. 2011. 25 Facts You Should Know About SAP. *Channel Insider* [online]. [vid. 16. březen 2013]. Dostupné z: <http://www.channelinsider.com/c/a/Spotlight/25-Facts-You-Should-Know-About-SAP-760260/>

KANGAS, Matt. 2005. *Introduction to the SAP Web Application Server*. březen 2005. S.l.: SAP.

KIM, W. Chan a Renée MAUBORGNE. 2005. *Blue ocean strategy: how to create uncontested market space and make the competition irrelevant*. Boston: Harvard Business School Press. ISBN 1-59139-619-0.

KOTLER, Philip a Kevin Lane KELLER. 2007. *Marketing management*. 12. vyd. Přel. Štěpánka ČERNÁ, Viktor FAKTOR a Tomáš JUPPA. Praha: Grada Publishing a.s. ISBN 978-80-247-1359-5.

KOUBEK, Ladislav. 2006. *Technologie v rámci mySAP ERP*. S.l. Diplomová práce. Vysoká škola ekonomická v Praze.

KÜHNHAUSER, Karl-Heinz. 2012. *ABAP: výukový kurz*. Dotisk prvního vydání. Přel. Ondřej BAŠE. Brno: Computer Press. ISBN 978-80-251-2117-7.

KUNSTOVÁ, Renata. 2009. *Efektivní správa dokumentů: co nabízí Enterprise Content Management*. 1. vydání. Praha: Grada Publishing. Management v informační společnosti. ISBN 978-80-247-3257-2.

PALTZER, Udo. 2010. *Available Adapters for SAP NetWeaver Process Integration 7.3* [online]. prosinec 2010. S.l.: SAP. Dostupné z: <http://scn.sap.com/docs/DOC-3500>

PAVLOVSKÝ, Roman. 2011. *Aplikace na správu projektů pro SAP NetWeaver Portal*. S.l. Diplomová práce. Vysoká škola ekonomická v Praze.

PITTARESE, Tony. 2012. Creating an undergraduate computing sequence focused on enterprise resource planning (ERP). *Journal of Computing Sciences in Colleges*. roč. 27, č. 5, s. 52–59 ISSN 1937-4771.

REJNKOVÁ, Petra. 2011. *Lokalizace a přizpůsobení metodiky OpenUP*. Praha. Vysoká škola ekonomická v Praze.

ŘEPA, Václav. 2007. *Podnikové procesy: procesní řízení a modelování*. 2. aktualiz. a rozš. vyd. Praha: Grada. Management v informační společnosti. ISBN 978-80-247-2252-8.

SAP. 2010. *Master Guide SAP NetWeaver 7.3*. 6. prosinec 2010. S.l.: SAP.

SAP. 2013a. ERP Solutions. *SAP* [online]. [vid. 5. březen 2013]. Dostupné z: <http://www54.sap.com/solutions/bp/erp/software/overview.html>

SAP. 2013b. Modeling Processes with Process Composer. *SAP Help Portal* [online]. [vid. 26. březen 2013]. Dostupné z: [http://help.sap.com/saphelp\\_nw73/helpdata/en/ce/19dc55105b46a0b498af9d840a93a8/frameaset.htm](http://help.sap.com/saphelp_nw73/helpdata/en/ce/19dc55105b46a0b498af9d840a93a8/frameaset.htm)

SAP. 2013c. Odvětví. *SAP Česká republika* [online]. [vid. 18. březen 2013]. Dostupné z: <http://www.sap.com/cz/industries/index.epx>

SAP. 2013d. SAP Business Management Software Solutions, Applications and Services. *SAP* [online]. [vid. 17. březen 2013]. Dostupné z: <http://www.sap.com/solutions/index.epx>

SAP. 2013e. SAP: lepší řízení firem [online]. [vid. 2. březen 2013]. Dostupné z: <http://www.sap.com/cz/about/index.epx>

SAP. 2013f. Zákazníci SAP ČR. *SAP Česká republika* [online]. [vid. 18. březen 2013]. Dostupné z: <http://www.sap.com/cz/about/customersuccess/index.epx>

SASKA, Slavomil. 2010. *Implementace ukázkové aplikace v kontextu SAP NetWeaver platformy* [online]. Praha [vid. 3. březen 2013]. Bakalářská práce. Vysoká škola finanční a správní. Dostupné z: <http://theses.cz/id/x4le3z/>

SODOMKA, Petr a Hana KLČOVÁ. 2012. Český ERP trh zrychlil růst, v segmentu SME přibýlo 2 000 projektů. *CVIS* [online]. [vid. 12. březen 2013]. Dostupné z: <http://www.cvis.cz/hlavni.php?stranka=novinky/clanek.php&id=1312>

TUNA, Gabriel. 2013. *Konzultace vývoje v systému SAP*. 23. leden 2013. S.l.: s.n.

TYLE, Jan. 2011. *Aplikace na správu projektů pro SAP NetWeaver Portal*. S.l. Diplomová práce. Vysoká škola ekonomická v Praze.

VOŘÍŠEK, Jiří. 1997. *Strategické řízení informačního systému a systémová integrace*. Praha: Management Press. ISBN 80-85943-40-9.

VOŘÍŠEK, Jiří. 2011. *Principy a modely řízení podnikové informatiky*. Praha: Oeconomica. ISBN 978-80-245-1440-6.

WOODS, Dan a Jeff WORD. 2004a. *SAP NetWeaver For Dummies*. Indianapolis, Indiana: Wiley Publishing, Inc. ISBN 0-7645-6883-3.

WOODS, Dan a Jeffrey WORD. 2004b. *SAP NetWeaver*. New Delhi: Wiley. ISBN 9788126505289.

# Seznam obrázků a tabulek

## Seznam obrázků

Obrázek 1: Zastoupení jednotlivých výrobců ERP systémů na českém trhu v segmentu velkých podniků .....	10
Obrázek 2: Evoluce podnikových aplikací společnosti SAP .....	11
Obrázek 3: Postavení na trhu společnosti SAP v oblasti ERP řešení pro středně velké podniky .....	12
Obrázek 4: Vztahy mezi komponentami, moduly, podnikovými scénáři a transakcemi .....	16
Obrázek 5: Schéma technologické architektury systému SAP .....	17
Obrázek 6: Schéma snižování TCO díky SAP NetWeaver .....	23
Obrázek 7: Schéma SAP Basis verze 4.6 a nižší .....	24
Obrázek 8: Schéma Web Application serveru .....	25
Obrázek 9: Schéma WebAS rozšířené o prostředí Java .....	26
Obrázek 10: Schéma platformy SAP NetWeaver .....	27
Obrázek 11: Schéma nástrojů SAP NetWeaver pro BPM .....	34
Obrázek 12: Hlavní obrazovka SAP WN AS ABAP .....	45
Obrázek 13: Správa objektů v nástroji ABAP Workbench .....	46
Obrázek 14: Rozložení centrálních klientů do 3 systémů SAP .....	48
Obrázek 15: Schéma transportu objektů mezi systémy .....	50
Obrázek 16: Komunikace mezi obrazovkovým procesorem a ABAP procesorem .....	51
Obrázek 17: Struktura logiky zpracování aplikačního programu .....	53
Obrázek 18: Schéma zpracování dynamického programu .....	54
Obrázek 19: Vztah mezi poli tabulky, datovými prvky a doménami .....	56
Obrázek 20: Grafické znázornění datového modelu „rezervace letenek“ .....	57
Obrázek 21: Zjednodušený datový model cvičného příkladu .....	59
Obrázek 22: Diagram případů užití ukázkové aplikace .....	61
Obrázek 23: Logika toku obrazovek a moduly volané v akcích PBO a PAI .....	63
Obrázek 24: Aplikace jako součást požadavku, který je připraven k odeslání .....	66
Obrázek 25: Uživatelské rozhraní ukázkové aplikace .....	67
Obrázek 26: Kontrola názvu hostitelského počítače .....	83
Obrázek 27: Úvodní obrazovka instalačního průvodce .....	84
Obrázek 28: Výběr adresáře pro instalaci systému .....	84
Obrázek 29: Nastavení hesla administrátora databáze .....	85
Obrázek 30: Spuštění aplikačního serveru a databáze .....	86
Obrázek 31: Přihlášení k aplikačnímu serveru přes webové rozhraní .....	87
Obrázek 32: Tvorba systémového přihlášení .....	88
Obrázek 33: Nastavení připojení k serveru .....	88
Obrázek 34: Úvodní obrazovka trial verze systému SAP .....	89
Obrázek 35: Informace o instalaci a licencích systému SAP .....	90
Obrázek 36: Počáteční obrazovka ABAB Dictionary .....	91
Obrázek 37: Založení datového prvku .....	92
Obrázek 38: Nastavení domény .....	93
Obrázek 39: Nastavení označení pro pole .....	93
Obrázek 40: Tabulka s nastavenými poli a datovými prvky .....	94
Obrázek 41: Technické nastavení tabulky .....	94
Obrázek 42: Výpis záznamů z databázové tabulky .....	95
Obrázek 43: První obrazovka nástroje ABAP Editor .....	96

Obrázek 44: Nastavení atributů prvního programu .....	97
Obrázek 45: Přiřazení programu do balíčku.....	97
Obrázek 46: Výsledek prvního programu.....	98

## Seznam tabulek

Tabulka 1: Nasycenost trhu s ERP produkty v ČR na straně poptávky.....	9
Tabulka 2: Údaje o obrazech společnosti SAP za jednotlivé kvartály roku 2012 .....	12
Tabulka 3: Porovnání řešení společnosti SAP pro malé a střední podniky .....	14
Tabulka 4: Mapování softwarových jednotek na případy užití.....	37
Tabulka 5: Funkční požadavky aplikace .....	60
Tabulka 6: Nefunkční požadavky aplikace .....	61
Tabulka 7: Definice aktérů pro případy užití .....	62
Tabulka 8: Specifikace případů užití .....	62

# Rejstřík

## — A —

ABAP, 5, 25, 28, 41  
  Editor, 43  
  Stack, 25  
  Workbench, 43, 58  
abstrakt  
  anglicky, iii  
  česky, ii  
adaptér, 29  
  specifický pro aplikaci, 30  
  specifický pro odvětví, 30  
  technický, 30  
akceptační testy, 42  
ALV, 55, 63  
API, 21  
aplikační programy, 50  
  dynamické programy, 50  
  reporty, 50  
aplikační vrstva, 17  
ASAP, 40  
Asseco Solutions, 9

## — B —

balíček, 46, 49  
bloky událostí, 52  
bloky zpracování, 52  
Blueprint, 42  
BPMN, 33

## — C —

CAF, 27  
cloud computing, 19  
CRM, 7

## — D —

Data modeler, 56  
databázová vrstva, 17  
datová základna, 7  
Datový prvek, 56  
Datový slovník, 55  
dialogové moduly, 52  
Dialogové programování, 53  
Dispečer, 44  
Doména, 56  
dynpro, 50

## — E —

ERP, 7  
  ERP II, 7, 13

## — F —

funkční specifikace, 42

## — H —

Helios Orange, 9

## — I —

IFRS, 12  
informační systém, 4  
integrace, 21  
  integrační broker, 21  
  point-to-point, 21  
ITS, 24

## — J —

Java, 5, 19, 25, 28, 41  
  Java EE, 41  
  Stack, 25  
jmenné konvence, 47  
JSP, 25

## — K —

klient, 16  
  klient/server, 16  
Klient, 47  
  Customizing and Development Client, 47  
  Production Client, 47  
  Quality Assurance Client, 47

## — M —

Middleware, 29  
MMSP, ii, iii, 3, 59, 60, 66, 68, 71  
  Fáze konstrukce, 64  
  Fáze rozpracování, 62  
  Fáze zahájení, 60  
  Fáze zavedení, 66  
MVC, 32  
mySAP, 10

## — O —

objekty, 46  
  lokální, 46  
odvětvové řešení, 17  
OLTP, 10  
on-demand, 8

## — P —

poděkování, iii  
podnikový scénář, 15

Pohled, 55  
Portál, 31  
pracovní proces, 44  
Pracovní proces  
    dialogů, 44  
    probíhající v pozadí, 44  
    řadící, 44  
    tiskový, 44  
prezentační vrstva, 17  
procedury, 52  
    funkční moduly, 52  
    metody, 52  
    podrutiny, 52  
procesor, 51  
    obrazovkový procesor, 51  
    procesor ABAP, 51  
Process Composer, 33  
Process Desk, 34  
Process Server, 34  
produkční systém, 42  
prohlášení, ii

## — R —

RFC, 24  
RFP, 42

## — S —

SAP, 4  
    Application Server, 24  
    Basis, 17  
    Business All-In-One, 14  
    Business By Design, 14  
    Business One, 14  
    Business Suite, 4, 13  
    CRM, 14  
    Enterprise SOA, 19, 20  
    ERP, 5, 7  
    GUI, 24, 36, 50  
    Historie, 10  
    NetWeaver, 5, 21  
    PLM, 14  
    R/2, 10  
    R/3, 6, 10, 70  
    SCM, 15  
    Společnost, 10  
    SRM, 15  
SAP ERP  
    Corporate Services, 15  
    Financials, 15  
    Humant Capital Management, 15  
    Operations, 15  
SAP mini, 58  
SAP NetWeaver  
    Application Server, 28, 43

Business Process Management, 33  
Business Warehouse, 31  
Business Warehouse Accelerator, 31  
Composition Environment, 32, 42  
Developer Studio, 32  
Enterprise Search, 32  
Identity Management, 28  
Information Lifecycle Management, 31  
integrační platforma, 26  
Master Data Management, 30  
Master Guide, 34, 36  
Mobile, 32  
Portal, 31  
Process Integration, 29  
Solution Manager, 28  
technologický základ, 26  
Visual Composer, 33

SCM, 7

Screen Painter, 54, 64, 100

SOA, 18, 40

    služby, 23

Software As A Service, 8

Stavební bloky, 34

    Druhy užití, 35

    Klienti, 36

    mapování, 36

    Samostatné stroje, 35

systém, 48

    pro testování a vývoj, 48

    produkční, 49

    zajišťující kvalitu, 48

system landscape, 48

## — T —

tabulka, 55

    terminologický slovník, 9, 61

TCO, 8, 21

TOC, 22

transakce, 15, 45

transakční systémy, 7

Transportní systém, 47, 49

## — U —

uvolnění požadavku, 49

## — V —

vrstva, 43

    aplikační, 43

    databázová, 43

    prezentační, 43

## — W —

Walldorf, 4

Web Application Server, 19

Web Dynpro, 6, 25, 32, 33  
WebAS, 24  
WSDL, 19

### — X —

XML, 19

### — Z —

Zákaznický vývoj, 34, 36, 40  
Zámek, 55



# Příloha A: Instalace SAP NetWeaver 7.01 ABAP trial verze

Příloha popisuje instalaci zkušební verze systému SAP, často označované jako „SAP mini“. Zkušební verze slouží pouze pro výukové účely a žádná aplikace vytvořená v tomto prostředí nemůže být transportována<sup>31</sup> do produkčního systému.

## A.1 Úvod

Instalace zkušební verze systému SAP spočívá ve stažení a instalaci aplikačního severu SAP NetWeaver a databáze MaxDB<sup>32</sup>. Na webu *SAP Community Network*<sup>33</sup> v sekci *downloads* jsou k dispozici jednotlivé instalační soubory v různých verzích. Nejnovější verze **SAP NetWeaver Application Server ABAP 7.03 64-bit Trial** je určena pouze pro 64-bitový operační systém Windows 7. Tato příloha popisuje instalaci verze **7.01**, která je určena pro Windows XP Professional.

### A.1.1 Požadavky

Ještě před stažením instalace je třeba zkontrolovat, zda váš operační systém a hardware bude vyhovovat požadavkům na bezproblémový chod instalace i samotného serveru. Neměli splněni být pouze jeden z požadavků (instalační průvodce na ně neupozorní), pak nebude možné se po instalaci přihlásit k serveru. Zkontrolujte proto všechny následující parametry vašeho hardware, operačního systému a jeho nastavení:

- **OS:** Windows XP Professional (Service Pack 2 nebo 3)
- Název počítače nesmí přesahovat 13 znaků
- **Systém souborů:** NTFS
- **Prohlížeč:** Internet Explorer 6.0 a vyšší nebo Firefox 1.0 a vyšší
- **Operační paměť RAM:** 2GB
- **Procesor:** Intel Pentium III/1.1 GHz a vyšší

---

<sup>31</sup> Více o transportech viz kapitolu **Chyba! Nenalezen zdroj odkazů.**

<sup>32</sup> MaxDB je databáze, kterou vyvinula společnost SAP a používá ji jako základní databázi pro své produkty. Podporovány jsou samozřejmě i databáze např. od firmy Oracle nebo Microsoft.

<sup>33</sup> <http://scn.sap.com>

- **Hard disk:** 50 volného místa dočasně během instalace, trvale 36 GB
- **Rozlišení monitoru:** 1024x768 px

Nemáte-li k dispozici počítač s OS Windows XP, doporučuji vytvořit virtuální pracovní plochu např. pomocí nástroje *VMware Workstation*<sup>34</sup>. Na celou instalaci dále doporučuji vyhradit **4 hodiny času**. Proběhne-li vše bez problému, instalace by měla trvat 2-3 hodiny.

### A.1.2 Stažení a rozbalení instalace

Pro stažení instalačních souborů je třeba založit profil na SCN a doplnit osobní údaje. Po ověření e-mailové adresy je možné se přihlásit a přejít na stránku stahování, která je dostupná na adrese <http://www.sdn.sap.com/irj/scn/nw-downloads>. Zvolte odkaz **SAP NetWeaver Application Server ABAP 7.02 SP6 32-bit Trial Version**. Instalační balík se skládá ze dvou komprimovaných souborů ve formátu RAR. Každý soubor postupně označte a stáhněte. Po rozbalení prvního souboru by měla vzniknout jediná instalační složka.

## A.2 Instalace SAP konzole

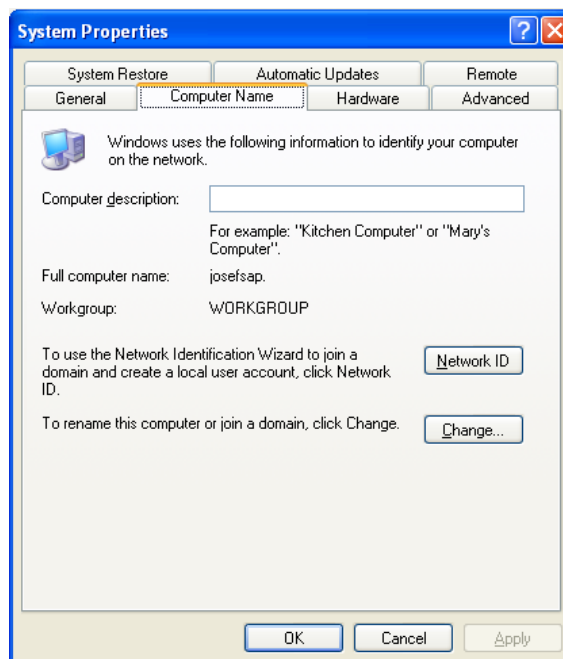
Prvním krokem je nainstalování programu *SAP Management Console*, který slouží ke správě systémů SAP a ovládání běhu aplikačního serveru. V instalační složce přejděte do adresáře **mmnc** a spusťte soubor **sapmmcX86u.msi**. Postupujte dle pokynů v průvodci a jako cestu, kam se má program instalovat, volte vždy takovou, která neobsahuje adresáře s mezerami.

## A.3 Instalace aplikačního serveru

Nejdelší fází je instalace aplikačního serveru. Na tomto místě je nezbytné se ujistit, že všechny požadavky uvedené v kapitole A.1.1, jsou splněny. Nejčastější chybou, což vyplývá z diskusních poradenských center, je nedodržení délky názvu hostitelského počítače. Proto ve Windows XP zvolte **Tento počítač → Zobrazit systémové informace → záložku Jméno počítače**. Pokud je v položce **Úplné jméno počítače** řetězec delší než 13 znaků, klikněte na tlačítko **Změnit**. Nový název se projeví až po restartování počítače.

---

<sup>34</sup> <http://www.vmware.com>



**Obrázek 26:**  
*Kontrola názvu hostitelského počítače*

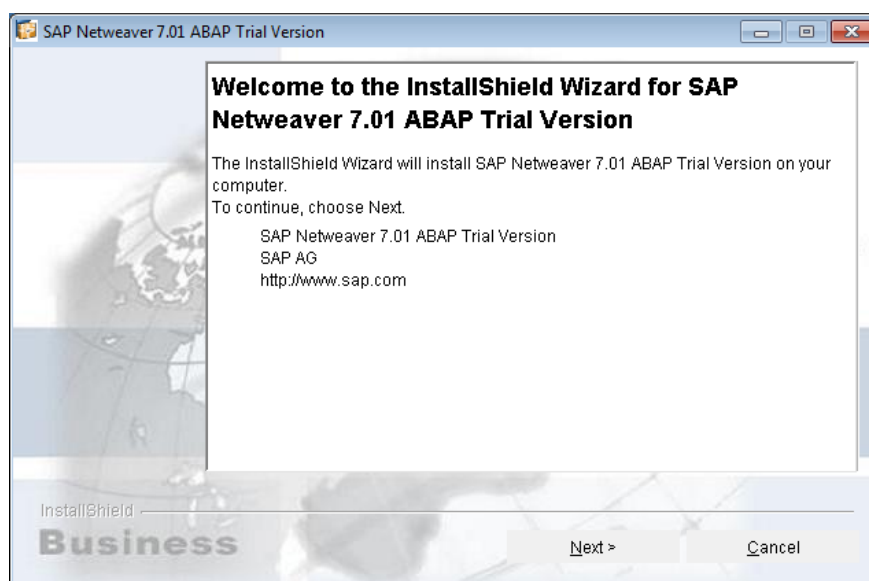
### A.3.1 Prerekvizity

Pro úspěšnou instalaci se předpokládají následující podmínky. Pokud vše souhlasí, vyhnete se tím opakování dvouhodinové instalace.

- Používáte Windows XP Service Pack 2 nebo 3.
- Program SAP Management Console je nainstalovaný.
- Hostitelský počítač má vyhovující počet znaků.
- Nejsou puštěné žádné další aplikace.
- Počítač je připojen k internetu.
- Na cílovém disku je více než 50 GB volného místa.
- Žádná další instalace systému SAP se v počítači nevyskytuje.

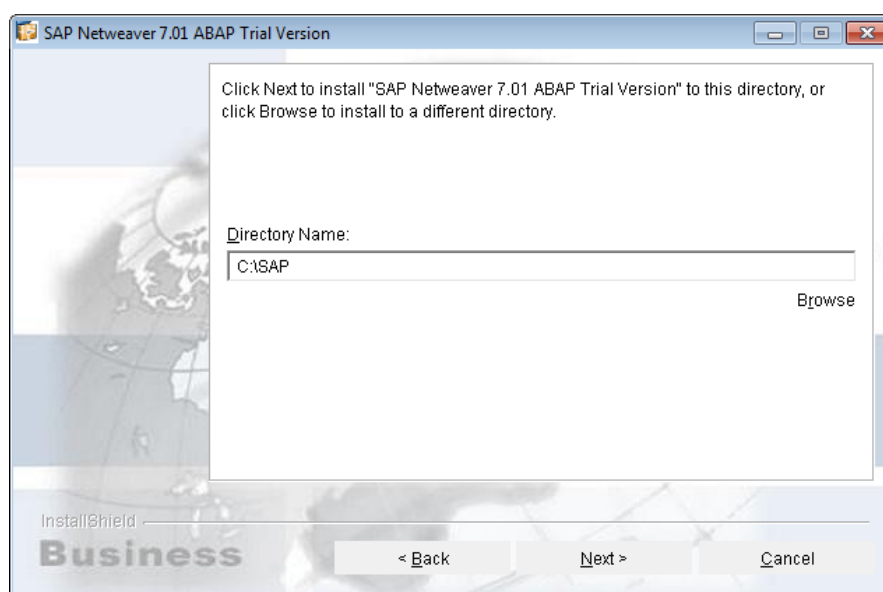
### A.3.2 Průvodce instalací

Nyní je již možné přistoupit k hlavní instalaci. Přejděte do adresáře **image** a spusťte soubor **setup.exe**. Postupujte dle pokynů instalačního průvodce.



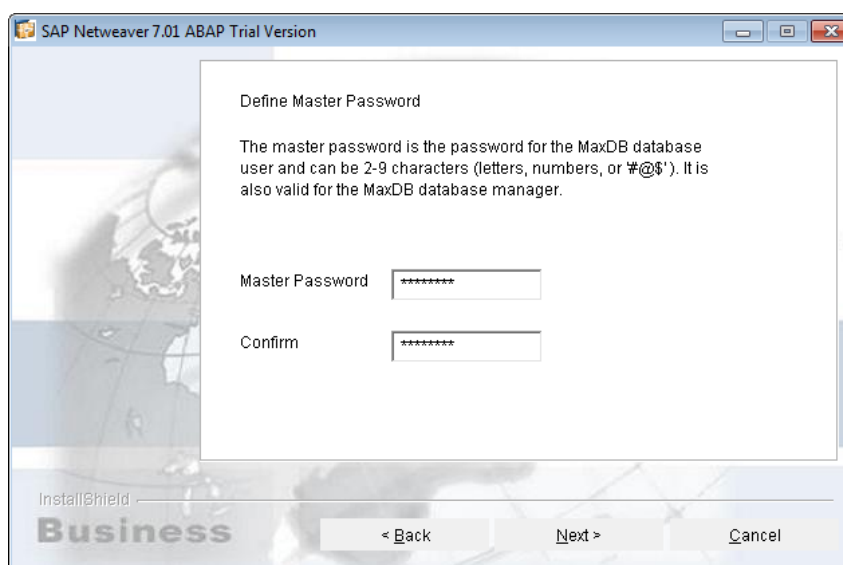
**Obrázek 27:**  
*Úvodní obrazovka instalačního průvodce*

Pokračujte odsouhlasením licenčních podmínek a zadejte cestu, kam se má server instalovat. Zde opět platí zadat cestu, která neobsahuje mezery v názvech adresářů.



**Obrázek 28:**  
*Výběr adresáře pro instalaci systému*

Posledním krokem je nastavení hesla administrátora databáze. Heslo musí být dlouhé 2-9 znaků, nesmí začínat číslicí a musí obsahovat kombinaci znaků abecedy i speciálních znaků.



**Obrázek 29:**  
*Nastavení hesla administrátora databáze*

Potom se zobrazí shrnutí předchozích kroků a instalace může začít. Klikněte na tlačítko **Install**. Během následujících 2 hodin (závisí na výkonu počítače) se nainstaluje aplikační server a databáze MaxDB. Po skončení instalace restartujte počítač.

### A.3.3 Kontrola

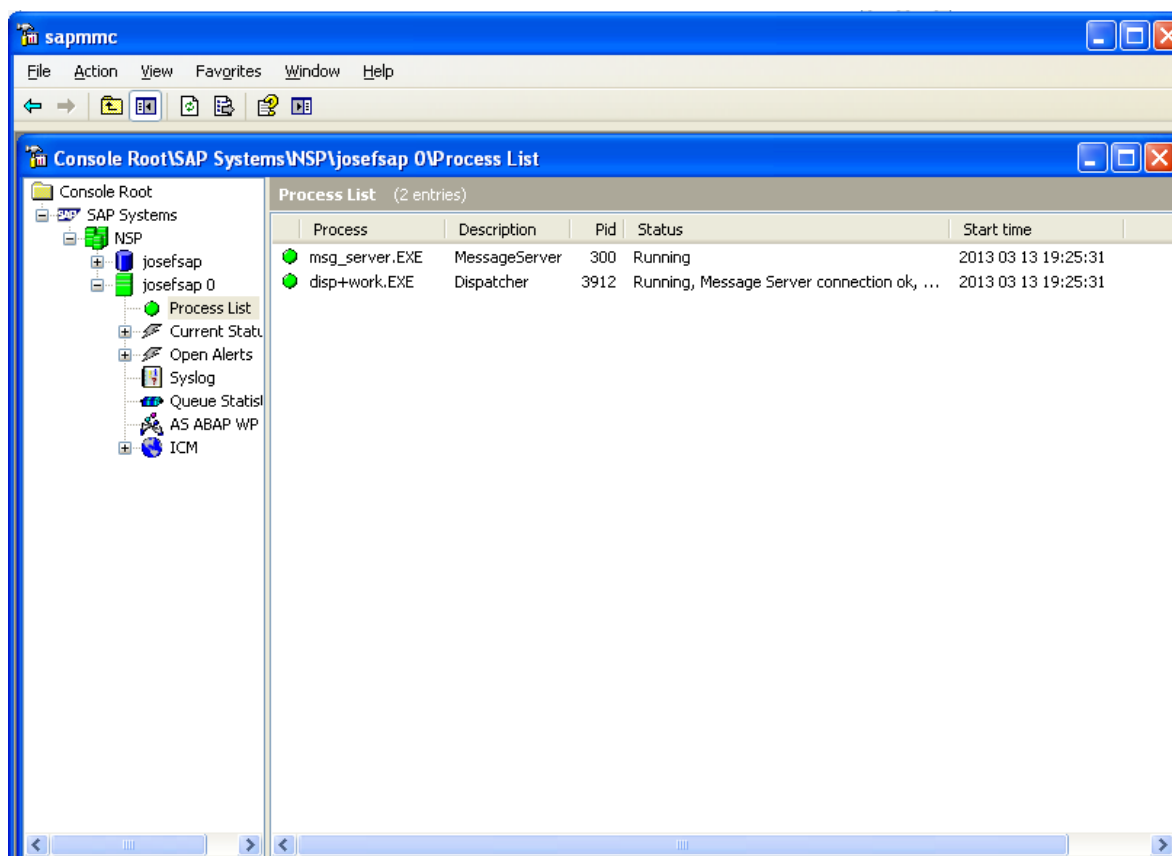
Před prvním přihlášením do systému a instalací dalších nástrojů doporučuji ověřit, zda všechno proběhlo v pořádku. Nejprve otestujte databázi. Otevřete příkazový řádek systému Windows a zadejte následující příkaz:

```
dbmcli -u control,<heslo> -d NSP db_state
```

<heslo> je administrátorské heslo databáze, které jste nastavili na začátku instalace. Měla by se objevit následující odpověď:

```
OK  
State  
ONLINE
```

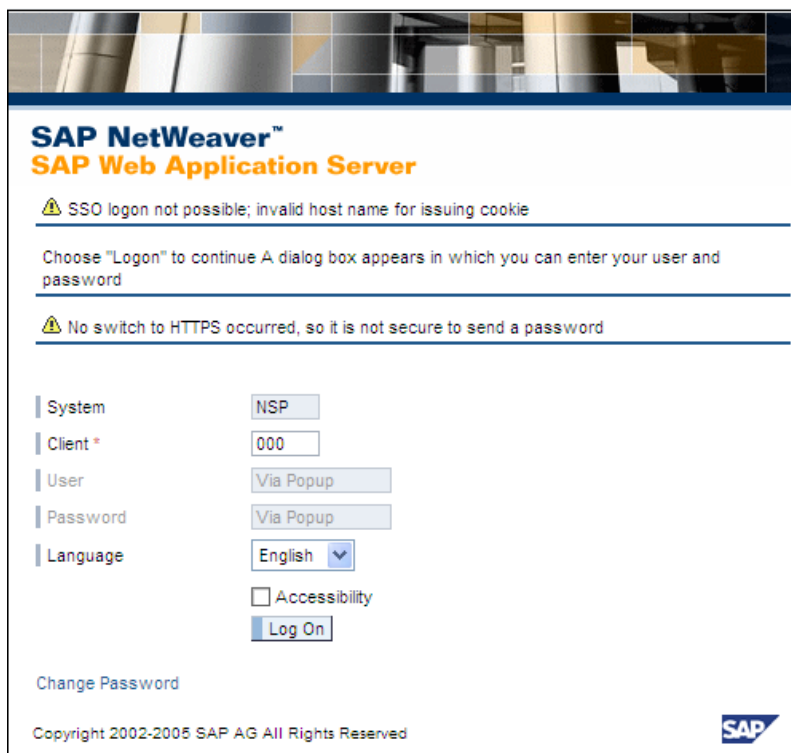
To znamená, že databáze byla úspěšně nainstalovaná, a je v online stavu. Nyní zkontrolujte, zda běží server a jeho procesy. Spusťte program *SAP Management Console* a ve stromové struktuře přejděte ze SAP Systems na NSP. Zde by měla být modrá ikona databáze a zelená ikona serveru. Nemí-li server ještě spuštěn, zvolte ve Windows nabídce Start → Všechny programy → **SAP NetWeaver Application Server 7.01 ABAP trial version** → NSP a klikněte na **Start Application Server**. Po návratu do konzole přejděte na položku **Process List** a vše by mělo vypadat tak, jak ukazuje Obrázek 30.



**Obrázek 30:**

*Spuštěný aplikační server a databáze*

Zejména se přesvědčte, že běží procesy `msg_server.EXE` a `disp+work.EXE`. Nyní zkontrolujte webové rozhraní. Otevřete webový prohlížeč a zadejte následující URL adresu: `http://localhost:8000/sap/bc/gui/sap/its/webgui?sap-client=000`. Zobrazí-li se HTML stránka s přihlášením k serveru, pak je vše v pořádku. Pro pohodlnější přihlášení je nyní možné nainstalovat nástroj **SAP Logon**, který využívá technologii **SAP GUI**.



**SAP NetWeaver™**  
**SAP Web Application Server**

⚠ SSO logon not possible; invalid host name for issuing cookie

Choose "Logon" to continue. A dialog box appears in which you can enter your user and password

⚠ No switch to HTTPS occurred, so it is not secure to send a password

System: NSP  
Client \*: 000  
User: Via Popup  
Password: Via Popup  
Language: English  
☐ Accessibility  
Log On

[Change Password](#)

Copyright 2002-2005 SAP AG All Rights Reserved

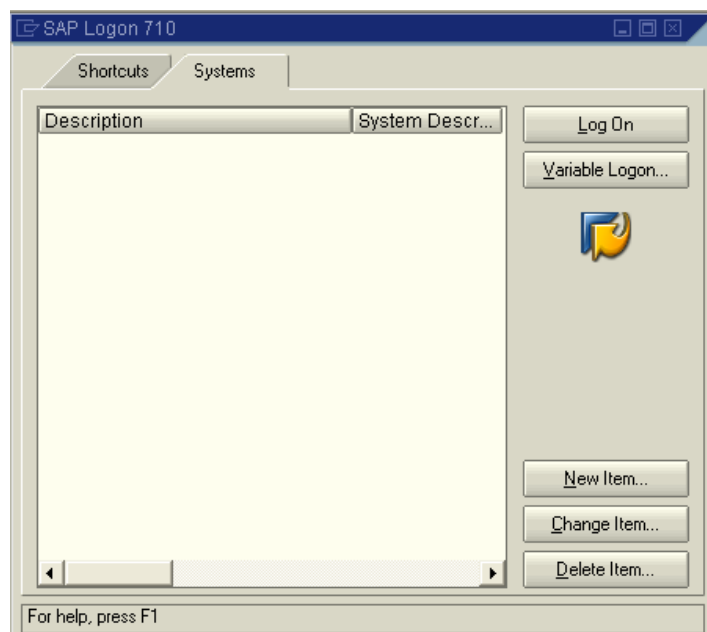
**Obrázek 31:**

*Přihlášení k aplikačnímu serveru přes webové rozhraní*

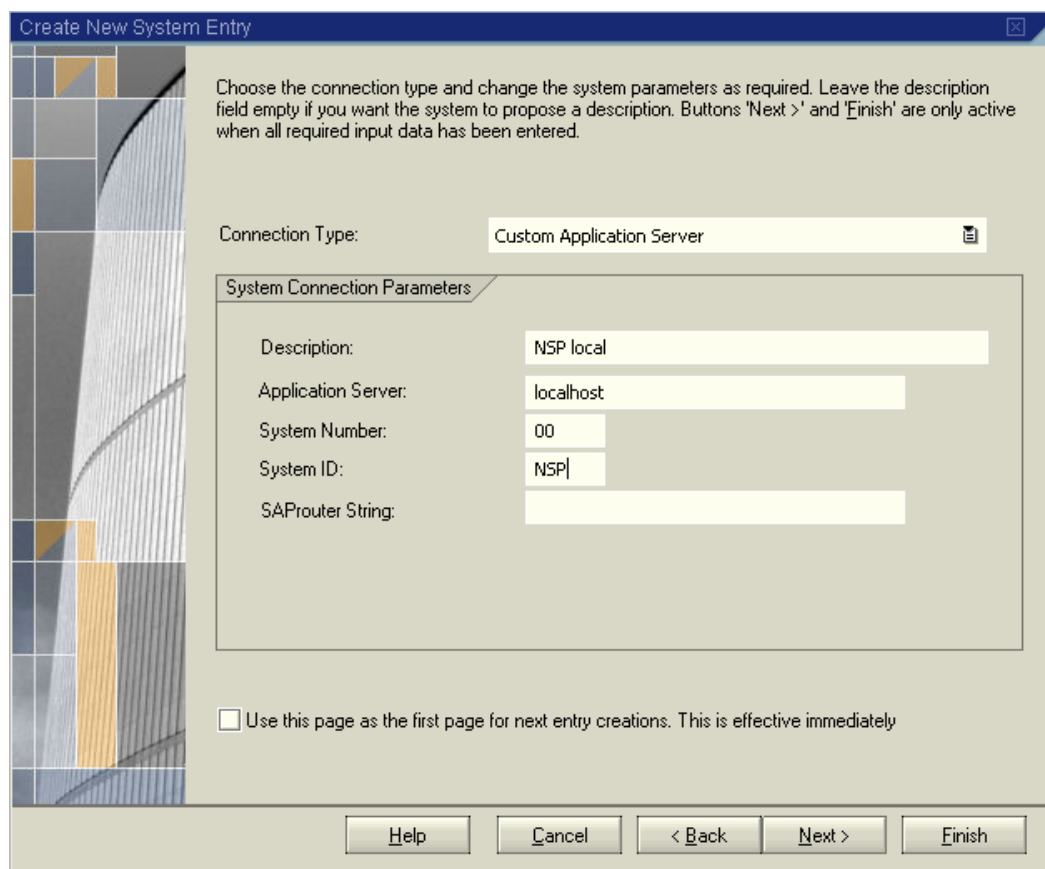
## A.4 Instalace a nastavení SAP Logon

Nyní můžete bez obav nainstalovat nástroj pro přihlášení do systému SAP přes grafické rozhraní SAP GUI. Nejprve **zastavte** aplikační server buď přes nabídku start, nebo pomocí SAP konzole. Přejděte do složky SAPGUI a spusťte soubor SapGuiSetup.exe. Postupujte dle pokynů průvodce a po dokončení instalace restartujte počítač. Spusťte ikonu nástroje SAP Logon, která se vytvořila na pracovní ploše. V záložce **Systems** klikněte na tlačítko **New Item**. Viz Obrázek 32.

V dalším okně vyberte **User Specified System** a pokračujte tlačítkem Next. Zobrazí se okno nastavení připojení k serveru. Vyplňte všechny hodnoty tak, jak ukazuje Obrázek 33 a klikněte na tlačítko **Finish**. Tím je vše připraveno pro první přihlášení.



**Obrázek 32:**  
*Tvorba systémového přihlášení*

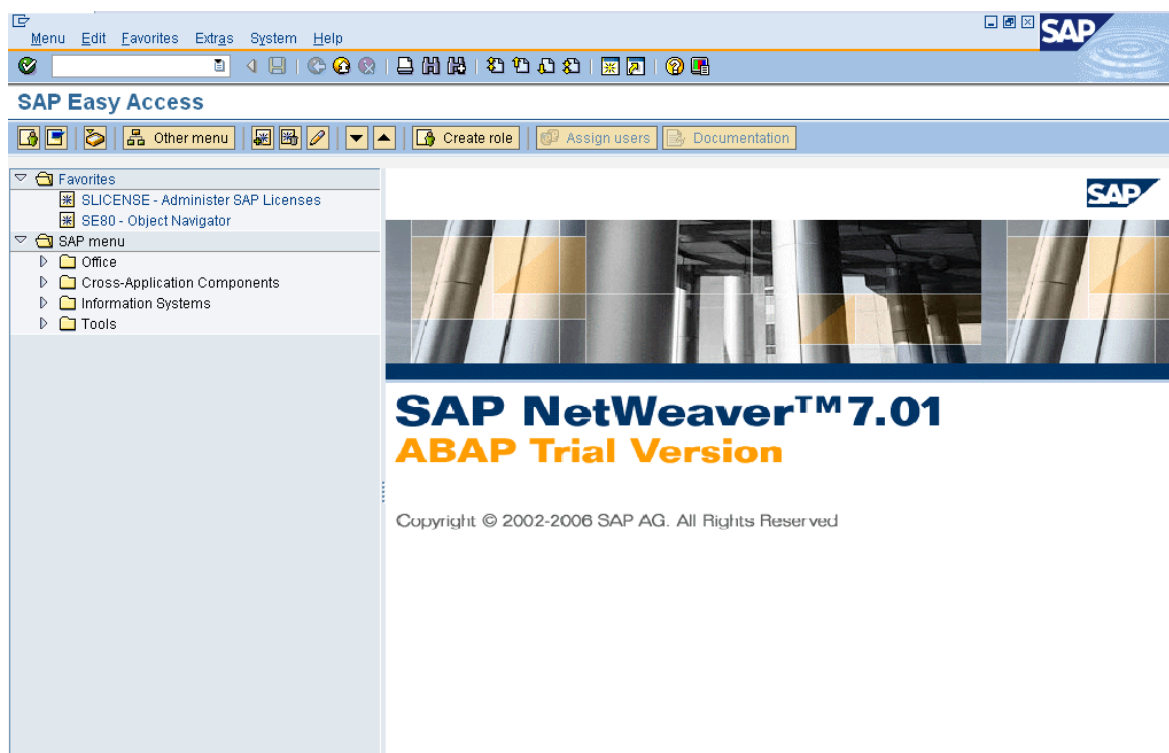


**Obrázek 33:**  
*Nastavení připojení k serveru*



### A.4.1 První přihlášení

Vraťte se na úvodní okno aplikace *SAP Logon* (Obrázek 32), zvolte položku systému, kterou jste pojmenovali *NSP local* a klikněte na tlačítko **Log On**. Zobrazí se přihlašovací okno systému SAP a jako vývojář napište do pole *username* hodnotu **BCUSER**. Heslo je vždy **minisap**. Tím je instalace dokončena a jste úspěšně přihlášení do systému SAP, jak ukazuje Obrázek 34.



**Obrázek 34:**

*Úvodní obrazovka trial verze systému SAP*

## A.5 Prodloužení licence

Po dokončení instalace začíná standardní třicetidenní lhůta, po jejímž uplynutí již není možné se do systému přihlásit. Společnost SAP však nabízí ještě bezplatné prodloužení licence pro všechny uživatele SCN.

Pro získání licenčního klíče je potřeba znát své identifikační číslo, které jste získali při registraci. Najdete ho v úvodním emailu, který jste obdrželi při zakládání účtu a je označeno jako **username**. Dále potřebujete znát hardware klíč nainstalovaného systému (*hardware key*). Přihlaste se do systému SAP pod přihlašovacím jménem **SAP\*** a heslem **minisap**. Zadejte transakci s názvem **sllicense** a potvrďte. Zobrazí se informace o instalaci

a licencích, jak ukazuje Obrázek 35. Hardwarový klíč najdete jako první položku na kartě **Current Settings**.

**SAP License Administration # Digitally-Signed License Keys**

Online Documentation | Old Licenses

---

**Current Settings**

Active Hardware Key:	P0895491781
Installation Number	DEMOSYSTEM
License Expiration Date	09.07.2013
System Number	00000000311929235

---

**Installed Licenses**

Licenses in the Database

Stat...	SID	Hardware Key	Sware Prod	Valid From	Valid To	Type	Inst. No.
○○	NSP	P0895491781	NetWeaver_ADA	09.04.2013	09.07.2013	Perm	DEMOS

Install

**Obrázek 35:**

*Informace o instalaci a licencích systému SAP*

Poté přejděte do webového prohlížeče a zadejte adresu <http://www.sap.com/minisap>, kde se nachází formulář s žádostí o zaslání licenčního klíče. Zde je nápověda pro některá pole:

- jako **SDN User ID** zadejte své identifikační číslo ze SCN,
- jako **System ID** zvolte z rozbalovací nabídky hodnotu **NSP**,
- jako **Hardware Key** zadejte svůj hardwarový klíč (viz Obrázek 35).

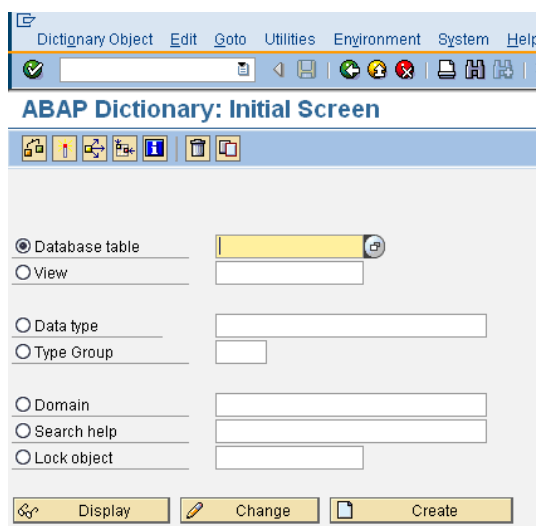
Poté odešlete formulář a po několika hodinách vám přijde na email soubor s licenčním klíčem. Přejděte opět do systému SAP a na kartě **Installed Licenses** klikněte na tlačítko **Install**. Vyberte soubor a potvrďte. Tím je licence prodloužena.

## Příloha B: Tvorba databázové tabulky v nástroji ABAP Dictionary

V této příloze popisují, jak vytvořit databázovou tabulku a některé datové prvky v nástroji ABAP Dictionary.

### B.1 Tvorba tabulky

Po přihlášení do systému (viz příloha A.4.1) spusťte nástroj ABAP Dictionary, a to buď zadáním transakce **SE11**, nebo projděte stromovou strukturou v MENU SAP. Objeví se hned nabídka pro vytvoření určitého objektu slovníku. Vyberte položku **Database table**, zadejte název tabulky a potvrďte tlačítkem **Create**. Tabulku nazvěte ZMEMBER. Tabulka bude obsahovat záznamy o členech vývojového týmu.



**Obrázek 36:**

*Počáteční obrazovka ABAP Dictionary*

Nejprve doplňte krátký popis tabulky.

Na kartě **Delivery and Maintenance** napište do pole **Delivery Class** písmeno A, což znamená, že se v tabulce budou vyskytovat kmenová data. Pod tímto polem ještě nastavte oprávnění pro úpravy tabulky na **Display/Maintenance allowed**.

Nyní tabulku uložte (Ctrl+s) a objeví se okno, kde je třeba vybrat balíček, do kterého bude příslušet tato tabulka. Protože se nepředpokládá, že bude tato výuková tabulka transportována, stiskněte tlačítko **Local object**. Tabulka je nyní dočasně vytvořena (má příznak *New*), ale není aktivována (nemá fyzickou reprezentaci v databázi). Proto musí obsahovat nějaké sloupce, neboli pole (*Fields*).

Na kartě **Fields** proto vytvořte první pole, které bude označovat aktuálního klienta a nazvěte ho CLIENT. Označte ho jako klíčové a jako datový prvek použijte MANDT, který už je vytvořen, a tak se automaticky vyplní jeho atributy a související domény. Pro další pole už ale musíte datové prvky vytvořit a případně jim přiřadit domény.

## B.2 Tvorba datového prvku a domény

Dalším polem bude číslo člena týmu, které nazvěte MNUMBER a označte ho také jako klíčové. Pro toto pole napište název datového prvku ZMNUMBER, který ale ještě neexistuje. Využitím tzv. „*dopředné navigace*“ klikněte na název datového prvku a objeví se výzva k založení datového prvku (viz Obrázek 37).

The screenshot shows the 'Data Type' tab in the ABAB Dictionary tool. At the top, the 'Data element' is 'ZMNUMBER' and it is marked as 'Active'. Below it, the 'Short Description' is 'Data element for number of member'. The 'Data Type' tab is selected, showing options for 'Elementary Type', 'Domain', 'Predefined Type', 'Reference Type', and 'Reference to Predefined Type'. The 'Domain' option is selected, and a new domain 'ZMNUMBER' is being created. The 'Data Type' is 'NUMC' (Character string with only digits), 'Length' is '5', and 'Decimal Places' is '0'. The 'Predefined Type' option is also visible with its own fields for 'Data Type', 'Length', and 'Decimal Places'.

**Obrázek 37:**  
Založení datového prvku

Vyplňte popis datového prvku a napište název domény se stejným názvem, jako je datový prvek – ZMNUMBER. Kliknutím na doménu se objeví okno pro její vytvoření. Zadejte popis a datový typ (NUMC – čísla a znaky) a rozsah hodnot (1 – 99 999). Nyní doménu uložte (Ctrl+s) opět jako lokální objekt – tím se automaticky vyplní údaje na kartě **Properties**. Pak je nutné doménu aktivovat (klávesa Ctrl+F3). Zobrazí se seznam dosud neaktivovaných objektů. Vyberte doménu a potvrďte aktivaci. Před aktivací se však doporučuje (stejně jako jakýkoliv jiný objekt) zkontrolovat, zda nevznikly nějaké nekonzistence. Kontrolu je možno provést stiskem Ctrl+F2.

Domain	ZNUMBER	Active
Short Description	Domain for number of a member	

Properties

Definition

Value Range

Format

Data Type: NUMC (Character string with only digits)  
No. Characters: 5  
Decimal Places: 0

Output Characteristics

Output Length: 5  
Convers. Routine:   
☐ Sign  
☐ Lower Case

**Obrázek 38:**  
*Nastavení domény*

Zpět k datovému prvku se vraťte pomocí klávesy F3. Nyní zbývá nastavit ještě označení pole na poslední kartě. Zde zvolte pro každou délku odpovídající označení, jak ukazuje následující obrázek.

Data element	ZNUMBER	Active
Short Description	Data element for number of member	

Attributes

Data Type

Further Characteristics

Field Label

	Length	Field Label
Short	3	MNO
Medium	7	MNumber
Long	13	Member_Number
Heading	3	MNo

**Obrázek 39:**  
*Nastavení označení pro pole*

Datový prvek je nyní možné aktivovat a používat. Tím je vytvořeno a nastaveno první pole tabulky. Obdobným stylem vytvořte další pole tabulky. Pro každé pole nový datový prvek, který však není třeba opatřovat doménou. Stačí, když zvolíte předdefinovaný typ dle svého uvážení – např. pro první jméno člena datový typ CHAR délky 50. Na konci by měla tabulka vypadat, jako na následujícím obrázku.

Field	Key	Init...	Data element	Data Ty...	Length	Deci...	Short Description
CLIENT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	MANDT	CLNT	3	0	Client
MNUMBER	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	ZMNUMBER	NUMC	5	0	Data element for number of member
MNAME	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	ZMNAME	CHAR	50	0	Name of a member
MSNAME	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	ZMSNAME	CHAR	50	0	Second name of a member
MDOB	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	ZMDOB	DATS	8	0	Date of birth of a member

Obrázek 40:

Tabulka s nastavenými poli a datovými prvky

### B.3 Dokončení tabulky

Jakmile dokončíte tvorbu polí, je tabulka připravena pro aktivaci. Předtím ale nastavte potřebná technická nastavení kliknutím na tlačítko **Technical settings** (viz Obrázek 41). Zde nastavte pole **Data Class** na hodnotu **APPL0** (kmenová data) a pole **Size category** na hodnotu 0, což určuje, že v tabulce bude maximálně 17 000 datových záznamů. Poté se vraťte zpět k tabulce (klávesa F3) a stiskněte tlačítko pro aktivaci tabulky nebo zkratku Ctrl+F3.

Name: ZMEMBER01 Transparent Table

Shorttext: Table of members

Last Change: BCUSER 01.04.2013

Status: New Not saved

**Logical storage parameters**

Data class: APPL0

Size category: 0

**Buffering**

☒ Buffering not allowed

☐ Buffering allowed but switched off

☐ Buffering switched on

**Buffering type**

☐ Single records buff.

☐ Generic Area Buffered

☐ Fully Buffered

No. of key fields: 1

☐ Log data changes

☐ Write access only with JAVA


Obrázek 41:

Technické nastavení tabulky

Pokud vše proběhlo v pořádku, můžete tabulku naplnit daty. Použijte hlavní menu, kde zvolte položku **Utilities**, dále **Table contents** a nakonec **Create entries**. Zde postupně zadávejte čísla, jména, příjmení a data narození členů vývojového týmu. Pole pro klienta nelze vyplnit, protože je dáno aktuálním klientem. Když dokončíte naplňování daty, můžete si tabulku zobrazit opět přes položku hlavního menu **Utilities**, **Table contents** a **Display**.

**table.** Objeví se výběr kritérií pro zobrazení záznamů. Chcete-li zobrazit všechny, stiskněte tlačítko **Execute**. Výsledek může vypadat jako na následujícím obrázku.

**Data Browser: Table ZMEMBER01 Select Entries 4**

					
Table: ZMEMBER01					
Displayed Fields: 5 of 5 Fixed Columns: 2 List Width 0250					
	CLIENT	MNUMBER	MNAME	MSNAME	MDOB
<input type="checkbox"/>	000	00001	JOSEF	RYBICKA	09.12.1988
<input type="checkbox"/>	000	00002	JAN	NOVAK	01.08.1960
<input type="checkbox"/>	000	00003	PETR	NOVOTNY	01.01.1958
<input type="checkbox"/>	000	00004	PAVEL	POULICEK	08.03.1973

**Obrázek 42:**

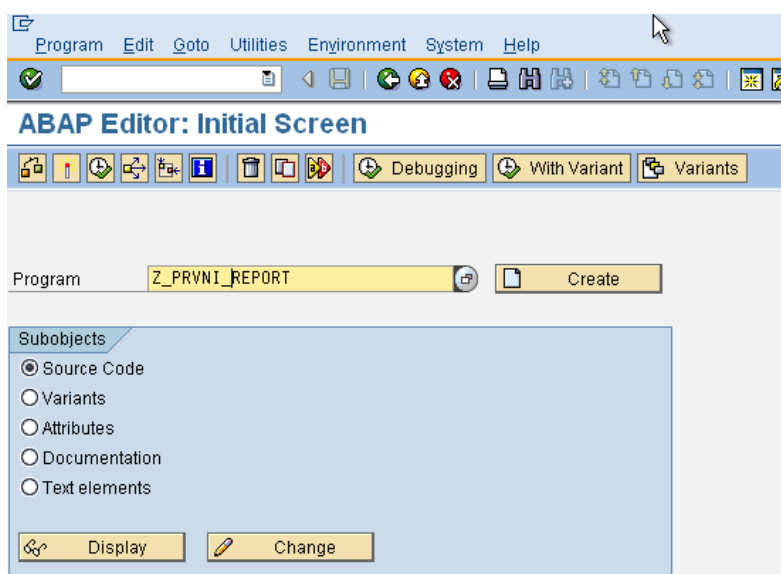
*Výpis záznamů z databázové tabulky*

## Příloha C: Vytvoření prvního programu

Tato příloha popisuje tvorbu prvního spustitelného programu ve zkušební verzi systému SAP v nástroji ABAP Editor.

### C.1 Vytvoření objektu

Přejděte do nástroje ABAP Editor pomocí transakce **SE38**. Jako první se hned objeví obrazovka pro založení programu, jak ukazuje Obrázek 43. Zadejte název programu a stiskněte tlačítko **Create**.



**Obrázek 43:**

*První obrazovka nástroje ABAP Editor*

Na dalším okně vyplňte titulek a jako typ programu zvolte **Executable program** (viz Obrázek 44:). Pokračujte stiskem tlačítka **Save**. Objeví se výzva pro přiřazení programu do balíčku. Program bude sloužit pouze pro výukové účely, a tak klikněte na tlačítko **Local object**. Tím se program automaticky přiřadí do balíčku \$TMP, který nelze transportovat a je zobrazen pouze uživateli, který ho vytvořil.



**Obrázek 44:**  
Nastavení atributů prvního programu

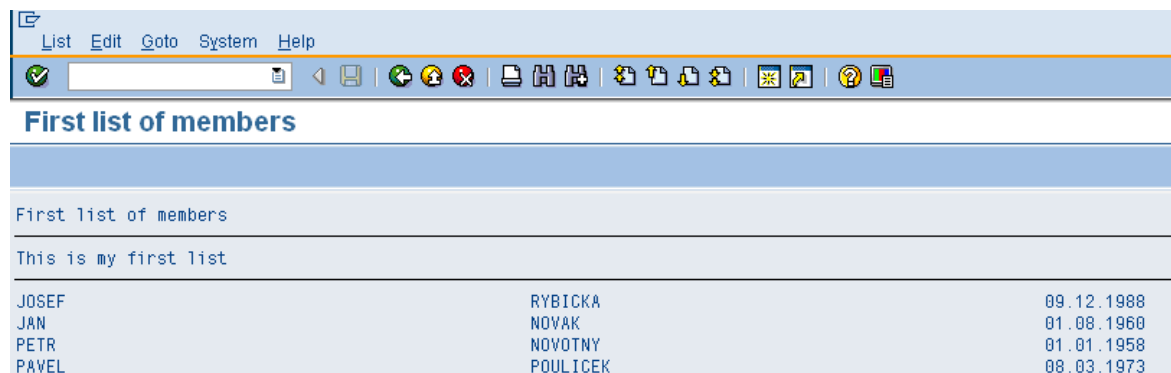
**Obrázek 45:**  
Přiřazení programu do balíčku

Tím je program vytvořen a spustí se obrazovka, která slouží pro editaci programu. Program bude vypisovat jména členů tabulky, kterou jste vytvořili v příloze C. Napište do editoru následující kód:

```
REPORT z_prvni_report.
TABLES zmember01.
WRITE 'This is my first list'.
ULINE.
SELECT * FROM zmember01.
WRITE: / zmember01-mname,
        zmember01-msname,
```

```
zmember01-mdob.  
ENDSELECT.
```

Poté uložte program (Ctrl+S), zkontrolujte syntaxi (F2) a pokud je program bez chyby, aktivujte ho klávesou (Ctrl+F3). Nyní můžete program spustit (F8) a mělo by se zobrazit to, co na následujícím obrázku.



**Obrázek 46:**  
*Výsledek prvního programu*

## Příloha D: Zdrojový kód ukázkové aplikace

V této příloze uvádím úplný zdrojový kód aplikace Z\_FLMANAGER.

```
*&-----*
*& Report  Z_FLMANAGER
*&
*& Program vypisuje letecke spoje. Po kliknuti na spoj,
*& se zobrazí lety, které jsou realizovány tímto spojením.
*&-----*
*& Autor    Josef Rybicka
*& Datum    4.4.2013
*& Verze    1.0
*&-----*

REPORT  z_flmanager.

*-----*
* CLASS event_handler DEFINITION
*-----*
* Definice tridy pro obsluhu udalosti
*-----*
CLASS cl_event_handler DEFINITION.
  PUBLIC SECTION.
    CLASS-METHODS: on_double_click FOR EVENT
if_salv_events_actions_table~double_click OF cl_salv_events_table
    IMPORTING
      row
      column.
ENDCLASS. "lcl_event_handler DEFINITION

*-----*
* Definice poli a datovych typu
*-----*

DATA:
  go_cont TYPE REF TO cl_gui_custom_container,
  go_cont2 TYPE REF TO cl_gui_custom_container,
  go_alv TYPE REF TO cl_salv_table,
  go_events TYPE REF TO cl_salv_events_table,
  go_functions TYPE REF TO cl_salv_functions_list,
  itab TYPE TABLE OF spfli,
  itab2 TYPE TABLE OF sflight,
  connection TYPE spfli-connid.

PARAMETERS: airline LIKE spfli-carrid,
             dep LIKE spfli-deptime.
```



```

*&-----*
*& Module init_0100 OUTPUT
*&-----*
* Zpracovani vystupu na obrazovku 0100
*-----*
MODULE init_0100 OUTPUT.
  SET PF-STATUS 'FLMNAV'.

  CHECK go_cont IS INITIAL.
  CREATE OBJECT go_cont
  EXPORTING
    container_name = 'VYPIS_SPOJU'.

  CALL METHOD cl_salv_table=>factory
  EXPORTING
    r_container   = go_cont
  IMPORTING
    r_salv_table = go_alv
  CHANGING
    t_table      = itab.

  go_events = go_alv->get_event( ).
  go_functions = go_alv->get_functions( ).

  go_functions->set_all( ).

  SET HANDLER cl_event_handler=>on_double_click FOR go_events.

  CALL METHOD go_alv->display.

ENDMODULE.                                "init_0100 OUTPUT

*-----*
* CLASS cl_event_handler IMPLEMENTATION
*-----*
* Telo tridy even_handler
*-----*
CLASS cl_event_handler IMPLEMENTATION.
  METHOD on_double_click.

    DATA: schedule TYPE spfli.

    READ TABLE itab INTO schedule INDEX row.
    IF sy-subrc = 0.
      connection = schedule-connid.
    ELSE.
      MESSAGE 'Nebyl nalezen zadny let' TYPE 'I'.
    ENDIF.

```

```

SET PARAMETER ID 'CON' FIELD connection.

CALL SCREEN 0200.

ENDMETHOD.                "on_double_click
ENDCLASS.                  "cl_event_handler IMPLEMENTATION

*&-----*
*&      Module  INIT_0200  OUTPUT
*&-----*
*      Zpracovani vystupu na obrazovku 0200.
*-----*
MODULE init_0200 OUTPUT.
  SET PF-STATUS 'FLMNAV'.

  GET PARAMETER ID 'CON' FIELD connection.
  SELECT * FROM sflight
  INTO TABLE itab2 WHERE connid = connection.

  CHECK go_cont2 IS INITIAL.
  CREATE OBJECT go_cont2
  EXPORTING
    container_name = 'VYPIS_LETU'.

  CALL METHOD cl_salv_table=>factory
  EXPORTING
    r_container   = go_cont2
  IMPORTING
    r_salv_table  = go_alv
  CHANGING
    t_table       = itab2.

  CALL METHOD go_alv->display.
ENDMODULE.                  "init_0200 OUTPUT

```

Aby kód fungoval, je třeba vytvořit příslušné obrazovky v nástroji *Screen Painter*, který je k dispozici v rámci ABAP Workbench. V Obrazovce 0100 je třeba vytvořit element s názvem VYPIS\_SPOJU typu custom container. Obdobně i v obrazovce 0200 vytvořte element s názvem VYPIS\_LETU. Logika toku pro obrazovku 0100 má následující kód:

```

PROCESS BEFORE OUTPUT.
MODULE init_0100.

```

```
PROCESS AFTER INPUT.  
MODULE user_command_0100.
```

Logika toku pro obrazovku 0200 má potom následující kód:

```
PROCESS BEFORE OUTPUT.  
MODULE init_0200.  
  
PROCESS AFTER INPUT.  
MODULE user_command_0200.
```