

# Prostředí pro integraci nástrojů manažera, vývojáře a programátora



Rozvoj informačních technologií s sebou přináší stále nové možnosti. Jednou z podstatných tendencí je i vývoj prostředků CASE, které prošly vývojem od čisté analytických nástrojů do podoby prostředků podporujících celý životní cyklus vývoje IS. Doplněním nástrojů CASE prostředky pro řízení procesů/projektů vzniká kvalitativně nové prostředí. Navíc tím, že je toto prostředí provozováno pod MS Windows nabízí dříve nepředstavitelný stupeň integrace. Jeho výhod mohou využívat všichni, kdo se na vývoji informačního systému podílejí.

- Prostředí pro integraci nástrojů manažera, vývojáře a programátora je nezbytným předpokladem pro to, aby bylo možné zvládnout a efektivně řídit proces vývoje velkých informačních systémů v architektuře klient/server.
- Často zmiňovaný pojem klient/server znamená, že aplikace je rozložena do dvou částí, klienta neboli front end a server neboli back end.
- Toto řešení umožňuje, aby na klientu byly používány moderní programovací nástroje, které zabezpečí flexibilitu řešení a zároveň i jeho nižší časovou a finanční náročnost oproti klasickým řešením.
- Na straně severu je možné využít spolehlivé a výkonné databázové stroje provozované na výkonných počítačích. Ve svém důsledku toto řešení vede k růstu spolehlivosti a flexibility řešení, což je u klasických řešení nemyslitelný požadavek.



## Problémy při vývoji aplikací

- vyčlenění značných lidských a finančních zdrojů
- velmi složité vztahy mezi objekty návrhu
- týmová práce
- velmi silná potřeba dokumentace s ohledem na další vývoj a zlepšování IS
- proces tvorby IS musí být říditelný
- je třeba zajistit opětovnou použitelnost vytvořeného kódu



2

**ILBMS**  
Česká společnost pro  
Informační Systémy

- Tvorba rozsáhlých aplikací v architektuře klient/server se nijak zásadně neliší od stavby nové továrny nebo od jiné, podobně složité činnosti. Vyžaduje dlouhodobé vyčlenění značných zdrojů, a to jak finančních, tak i lidských. Účastní se jí týmy vývojářů, které musí pracovat s tisíci komponent. Všechny činnosti musí být pečlivě zdokumentovány, aby bylo možné vyvinutý systém udržovat a dále zlepšovat.
- Nelze začít budovat továrnu bez toho, aby procesy realizující konečný produkt byly jasně a srozumitelně definovány. Stejně tak je nutné přistupovat k tvorbě IS. Musí zde existovat možnost efektivně řídit vývojový tým tak, aby bylo možné zjišťovat odchylky současného stavu od plánu, sledovat postup prací a zároveň detailně dokumentovat jednotlivé kroky. V opačném případě hrozí značné ztráty, které mohou být srovnatelné se zřícením špatně realizované stavby.
- V průmyslové praxi se při návrhu nového výrobku vychází ze dostupných standardizovaných komponent a navrhují se pouze ty části, které jsou jedinečné. Tento přístup je proto třeba uplatnit i v procesu tvorby IS. Je třeba využívat nástroje, které umožní opětovně používat již vytvořený kód. V procesu tvorby IS existuje velké množství vyzkoušených komponent, které ale nejsou přístupné, neboť neexistují mechanismy, jak je evidovat a pak opětovně použít. Tím se velmi zpomaluje proces tvorby funkčního IS.



## Požadavky na klient/server

- přizpůsobitelnost IS potřebné velikosti
- otevřenost
- řízení rizik
- vysoká produktivita práce



- Zvětšit rozsah informačního systému není v praxi jednoduchou záležitostí. Růst složitosti IS neprobíhá lineárně. Větší systém přináší problémy vyvolané složitostí vztahů mezi objekty. Je třeba řešit nepředvídané události, které s sebou růst velikosti IS přináší.
- V souvislosti s vývojem aplikací v architektuře klient/server je častým požadavkem otevřenost vývojového prostředí. Při detailnější analýze tohoto požadavku zjistíme, že jde spíše o úzkou integraci vývojových nástrojů. V praxi není potřebné, aby bylo možné používat co nejvíce vývojových nástrojů od různých dodavatelů, ale podstatné je, aby tyto nástroje uměly spolupracovat. Často dochází k tomu, že dodavatelé vývojových nástrojů, kteří proklamují otevřenost prostředí jsou ve skutečnosti nejméně otevření.
- Současně s růstem složitosti vývijeného systému roste nelineárně i riziko. Řízení rizik, které není klíčovým momentem v procesu tvorby malých systémů, se u velkých stává kritickou komponentou.
- Malé IS, vytvářené pomocí vizuálních nástrojů 4GL, bývají dokončeny v poměrně krátké době. Ne všechny zkušenosti však jsou použitelné pro velké systémy. Na důležitosti získává problém efektivního řízení vývojového týmu.



Výše popsané problémy nás vedou k úvaze, jaký vývojový postup použít.

Nejznámějším postupem je tzv. vodopád. Je detailní, k jeho záporným charakteristikám ale patří to, že snižuje produktivitu práce vývojářů a prodlužuje čas řešení projektu. Klasický postup zvyšuje nejistotu a může vést k chybným rozhodnutím o dalším postupu projektu. Jak vyplývá z jeho charakteristiky, není flexibilní, neboť konečný produkt je dodán ve zcela jiném čase, než bylo tvořeno zadání.

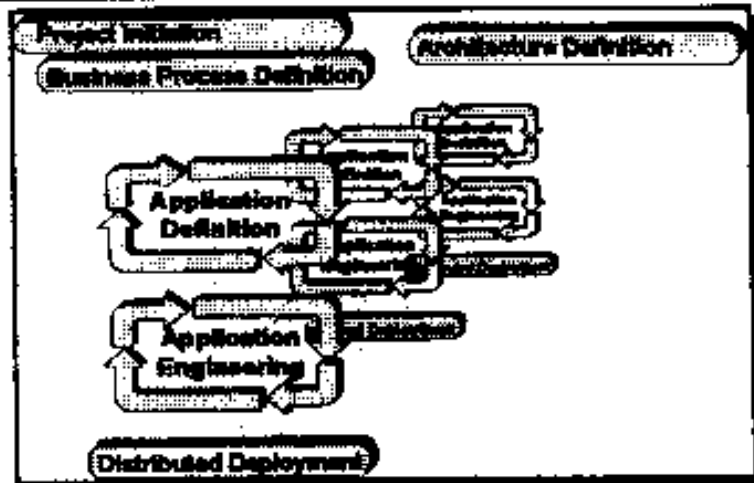
Postup "just do it", který zcela opomíjí fáze analýzy a návrhu dosahuje sice rychle výsledků, ale řešení není kompletní. Produktivita práce je vysoká, ale proces tvorby aplikace je zcela mimo kontrolu. Nejsou definovány žádné vývojové procesy, nelze opětovně používat již hotové objekty. Neumožňuje využít potenciálu 4GL nástrojů nabízejících možnost rychle vytvořit prototyp. Postupujeme-li při vývoji IS podle této strategie, získáme neudržovatelný IS. Tím, že chybí návrh, nejsou obvykle pokryty všechny požadavky uživatelů.

Objektově orientovaný návrh (OOD) neposkytuje podporu pro návrh s grafickým uživatelským rozhraním (GUI) ani pro návrh v architektuře klient/server. Otázkou zůstává, do jaké míry tento přístup poskytuje obraz o reálném světě. Největší problémy pak vznikají při implementaci objektového návrhu do prostředí relačních databází, neboť objektová databáze není v současnosti k dispozici.

Spirálový postup vývoje rozdělí aplikaci na říditelné přírůstkové fáze, ve kterých je možné opětovné použití objektů. Dále umožňuje využívání otevřeného prostředí pro vývojové nástroje a zároveň i obousměrnou komunikaci mezi navazujícími fázemi návrhu a implementace a jejich příslušnými nástroji.



## Spirálovitý postup vývoje



- Integruje prototypování a návrh
- Možnost opakovaného použití objektů

- Iterativní vývoj
- Nové techniky pro novou technologii

**Výsledek: Vysoký stupeň kontroly, kvalitní produkt.**

**LEMS**  
ČESKÁ REPUBLIKA

5

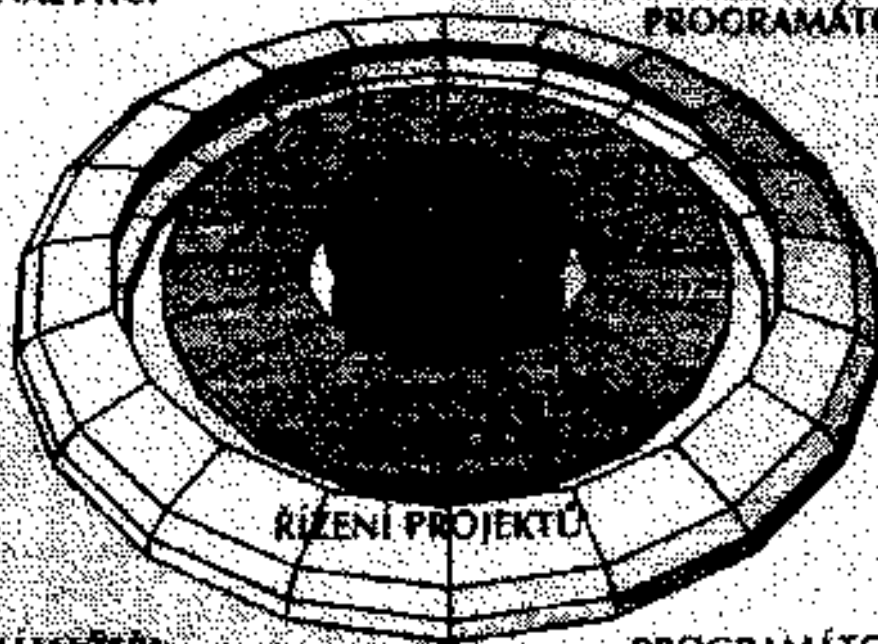
- Velké aplikace realizované v architektuře klient/server vyžadují zcela nový postup vývoje, neboť všechny výše uvedené postupy mají pro tuto architekturu nezanedbatelně záporné charakteristiky.
- Nový postup musí kombinovat výhody vysokého stupně kontroly klasického postupu (vodopád) a postupu rychlého vývoje aplikace (RAD) s využitím možností opětovného použití objektů a využít i kladné vlastnosti objektově orientovaného návrhu (OOD). Cílem je vytvořit procesy řízený postup, který připomíná montážní linku.
- Výsledná kvalita aplikace vytvořené na "montážní lince" závisí velmi silně na kvalitě předem připravených, opětovně použitelných objektů. Podstatnou vlastností těchto objektů je, že jsou dostatečně vyzkoušeny a zdokumentovány - zároveň musí být lehce přístupné všem vývojářům. Tento požadavek může zajistit pouze slovník dat. "Montáž" sama o sobě postupuje po jasně definovaných přírůstcích.
- Výsledný postup připomíná spirálu, která integruje úspěšné fáze prototypování a návrhu, které se opakují v iteracích. Spirálový postup vývoje rozdělí aplikaci na říditelné přírůstkové fáze, ve kterých je možné opětovné použití objektů.
- Spirálový postup uznává nutnost návrhu a také využívá i výhod prototypování. Síla vizuálních nástrojů 4GL svádí k tomu, aby byly přeskočeny fáze návrhu. Aplikace vyvinutá bez reálného návrhu má nekonzistentní rozhraní a je velmi těžko udržovatelná.



# Koncept CORE

ANALYTICI

ANALYTICI-  
PROGRAMÁTOŘI

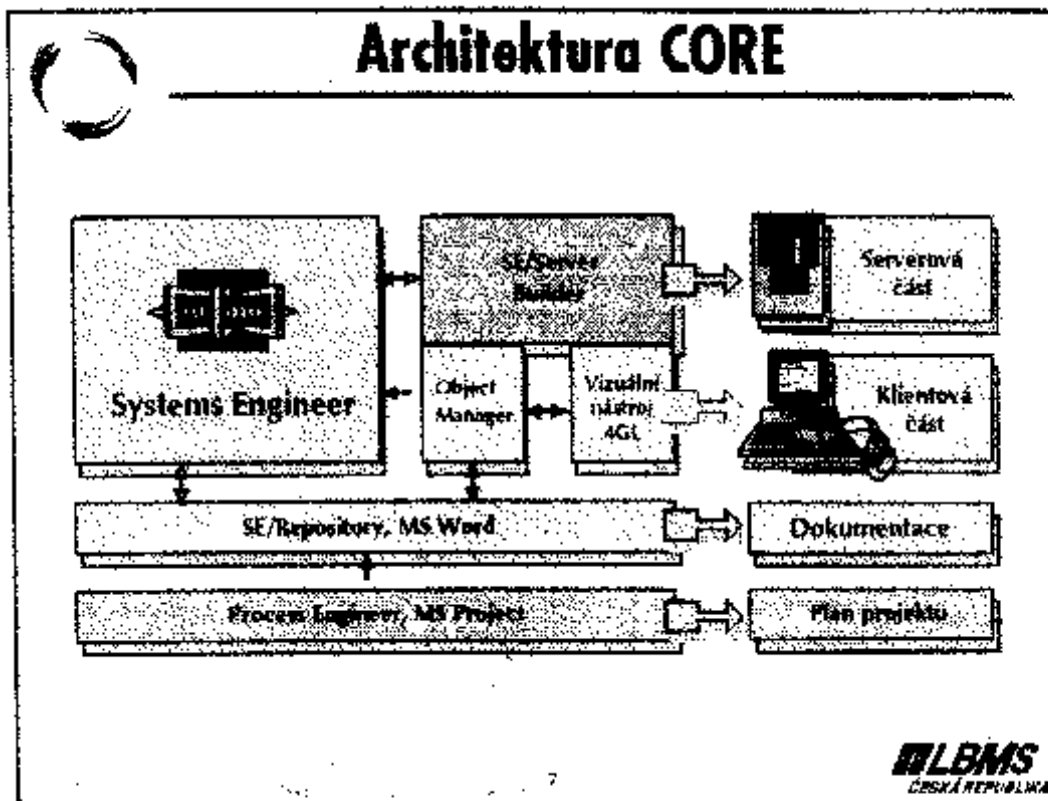


MANAŽERŮ

PROGRAMÁTOŘI

**LBMS**  
ČESKÁ REPUBLIKA

- Na základě předcházejících teoretických úvah navrhla firma LBMS koncept CORE.
- CORE (Common Object Repository Environment) je prostředí pro vývoj aplikací, které :
- umožňuje vytvořit heterogenní prostředí vývojových nástrojů, z nichž každý představuje špičku ve své třídě,
- je řízeno procesy,
- je vybudováno nad společnou objektovou repository,
- podporuje obousměrnou komunikaci mezi fázemi návrhu a implementace,
- poskytuje podporu pro práci manažerů, analytiků i programátorů.

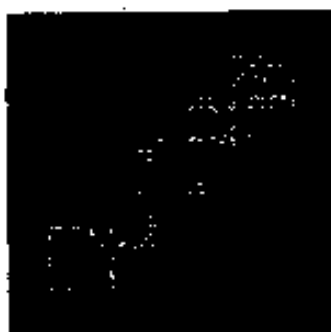


- CORE prakticky realizuje integrované prostředí pro podporu práce manažera, vývojáře i programátora.
- Manažer může využívat pro řízení procesů/projektů prostředek Process Engineer. Procesem se rozumí celkový technologický rámec, který umožní tvorbu konkrétních projektů. Např. v automobilce existují jasně definované procesy, které slouží pro realizaci různých projektů, např. sportovního vozu, luxusní limuzíny atd. Cílem v oblasti informačních technologií je do budoucna také tento přístup. Software by měl být "vyráběn".
- S pomocí prostředku Process Engineer stanovuje manažer úkoly jednotlivým analytikům a programátorům. Vazba tohoto prostředku na MS Project zabezpečuje, že pro samotné řízení projektu bude použito adekvátního (efektivního) nástroje. Pomocí tohoto prostředku je také možné provádět drobné změny plánu.
- Vývojář používá prostředek Systems Engineer pro analýzu a návrh IS. Výsledky své práce ukládá do objektové repository, kde jsou dostupné pro případné opělovné použití v rámci týmu vývojářů. Pro potřeby dokumentace má k dispozici služby repository v kombinaci s libovolným textovým editorem pro prostředí Windows.
- Databázový administrátor využije služeb modulu SE/Server Builder pro generování struktury databáze včetně uložených procedur a triggerů.
- Programátor grafického uživatelského rozhraní použije služby prostředku Object Manager k obousměrnému přenosu do a z repository CASE prostředku Systems Engineer. Otevřená koncepce objektové repository umožňuje uživateli zvolit nejvhodnější vývojový nástroj 4GL.



## Řízení projektů za pomoci CORE

- napomáhá efektivnímu řízení projektů
- podpora práce manažera
- zvyšuje produktivitu práce vývojářů
- podporuje opětovné použití objektů
- podporuje použití nástrojů pro rychlý vývoj aplikace
- umožňuje tvorbu vlastních postupů podle potřeb organizace



**LEMS**  
CORPUS SYSTÉMY

- Koncept CORE poskytuje všechny potřebné nástroje pro podporu řízení vývoje IS. Vrstva řízení procesů je realizována pomocí prostředku Process Engineer.
- Process Engineer umožní, že každý vývojový tým si může pružně vytvářet své vlastní postupy tvorby IS s využitím existující knihovny postupů (Process Library). K dispozici jsou postupy pro klient/server, rychlou dodávku systému (RAD), řízení projektů, strategické plánování, výběr softwarového balíku, klasický postup (vodopád), expresní postup, přírůstkový postup a postup pro údržbu. Konkrétní postup může vzniknout kombinací existujících postupů se specifickými aktivitami vyplývajícími z charakteristik daného projektu.
- Za pomoci prostředku Process Engineer může manažer navrhovat postup prací na projektu tak, aby přesně odpovídal charakteristice projektu. K dispozici má know-how ověřené na mnoha rozsáhlých projektech uložené v knihovně procesů.
- Process Engineer nabízí možnost definování jednotlivých rolí působících ve vývojovém týmu. Umožňuje plánování projektu s využitím různých formálních popisů a s výstupem do prostředí plánovacích nástrojů, jako je např. Microsoft Project.
- Po úspěšném ukončení projektu lze na základě zkušeností upravit projektové postupy (např. změnit odhady trvání jednotlivých etap, kroků a činností).





## CORE - vývojové nástroje

- možnost nasadit nástroje "best in class"
- obousměrná komunikace mezi fázemi návrhu a implementace
- objekty návrhu mohou být použity pro prototypování v prostředí vývojových nástrojů 4GL
- datový model lze použít pro vygenerování schématu relační databáze
- je možný re-engineering z relačních databází
- vývojové prostředí je flexibilní, lze měnit cílová prostředí
- CASE nástroj podporující vývoj je zcela uzpůsoben na tvorbu aplikací v architektuře klient/server

**LBMS**  
ČESKÁ REPUBLIKA

- Koncept CORE napomáhá tomu, že vývojáři mohou používat vývojové nástroje, které jsou nejlepší ve své třídě. Nejsou vázání na nástroje jedné firmy, ale mohou používat právě ty, které nejlépe vyhovují jejich potřebám.
- Obousměrný přenos mezi vizuálním programátorem a nástrojem CASE, je nezbytným předpokladem toho, aby mohl být používán spirálový postup vývoje IS.
- Objekty návrhu je možné přenášet z prostředí CASE do prostředí vývojových nástrojů 4GL, kde lze provádět prototypování. Zároveň lze využít datový model pro generování schématu relační databáze.
- Informace z prostředí vývojového nástroje 4GL lze přenášet zpětně do objektové repository. Je také možné provádět re-engineering existující relační databáze a vytvořit tak datový model.
- Otevřenost prostředí umožňuje, aby navržená aplikace byla realizována v kombinaci několika cílových prostředí. Aplikace vytvořená např. pro databázi Watcom může být reverzována a znovu vygenerována např. do databáze Sybase, možný je i opačný postup, rovněž seznam podporovaných databází je mnohem širší.



## CORE - objektová repository

- poskytuje integraci pro objekty vývojových nástrojů od různých dodavatelů
- umožňuje vytvářet a řídit množinu verzí jednotlivých objektů
- nabízí tvorbu sestav a analýzu dopadů
- lze do ní ukládat objekty vytvořené CASE nástrojem, databází, vývojovými nástroji 4GL i jinými nástroji
- uživatel může podle vlastních potřeb vytvářet verze jednotlivých objektů návrhu
- lze seskupovat jednotlivé verze objektů do tzv. základních sad (baseline) = configuration management
- zajišťuje opětovné použití objektů



**ILBMS**  
ČESKÁ REPUBLIKA

- Objektová repository poskytuje integraci pro objekty vývojových nástrojů od různých dodavatelů. Vývojový tým není striktně vázán na produkty pouze jedné firmy, v jejíchž silách není, aby všechny produkty, které dodává byly na špičkové úrovni. Vývojové nástroje firem specializovaných na tento typ produktu dosahují obecně lepších parametrů než nástroje firem, které se snaží dodávat celé spektrum software od databází až po vývojové nástroje 4GL.
- Možnost vytvářet a řídit množinu verzí jednotlivých objektů je klíčovou vlastností objektové repository. Verzování zamezuje tomu, aby si vývojáři vzájemně přepisovali výsledky své práce v případě týmové spolupráce.
- Z objektové repository lze na základě potřeb vývojářů vytvářet sestavy a řešit anlýzu dopadů, což je zvláště přínosné v případě změny návrhu.
- Seskupování jednotlivých verzí objektů do tzv. základních sad (baseline) zabezpečuje vývojářům, že budou pracovat vždy s konzistentní sadou objektů. Je možné pracovat s aktuální sadou objektů nebo obnovit libovolnou sadu tvořící dřívější základní sadu.
- Vývojáři mohou "montovat" jednotlivé aplikace z předem otestovaných a zdokumentovaných komponent a modifikovat je pouze v případě potřeby.



## Přínosy konceptu CORE

- poskytuje praktický přístup budování velkých aplikací v architektuře klient/server
- snižuje riziko nekontrolovatelného vývoje projektu IS
- podporuje celý životní cyklus projektu
- poskytuje prostředí pro integraci všech činností v procesu vývoje IS - od návrhu přes kódování až po řízení projektů



**LBMS**  
ČESKÁ REPUBLIKA

- Praktická realizace konceptu CORE dává manažerům, vývojářům i programátorům do rukou silný nástroj. Je zaměřena na praktické aspekty všech fází tvorby IS.
- Jasnými definováním činností, nástrojů i rolí zúčastněných v procesu tvorby IS minimalizuje riziko nekontrolovatelného vývoje projektu.
- Koncept CORE i jeho realizace poskytují podporu zúčastněným pracovníkům ve všech fázích životního cyklu projektu.
- Všechny činnosti, které je třeba v procesu tvorby IS realizovat najdou v konceptu CORE příslušný nástroj. Strategické plánování a řízení projektů podporuje Process Engineer, analýza, návrh i kódování naleznou podporu v prostředku Systems Engineer a souvisejících modulech SE/Server Builder a Object Manager.