

# **DIPLOMAMUNKA**

**Bende Mihály**

**Debrecen**

**2009**

Debreceni Egyetem  
Informatikai kar

# **Vállalatirányítási rendszerek kis- és középvállalati környezetben**

Témavezető:

Prof. Dr. Végh János

Egyetemi tanár

DE – Informatikai kar

Informatikai Rendszerek és Hálózatok Tanszék

Készítette:

Bende Mihály

Programtervező matematikus

Debrecen

2009

## Tartalomjegyzék:

<b>1.</b>	<b>Bevezetés</b> .....	5. oldal
<b>2.</b>	<b>Fogalmak</b> .....	7. oldal
<b>3.</b>	<b>Vállalatirányítási rendszerekről általánosan</b> .....	11. oldal
3.1	Információs rendszerek hőskora.....	11. oldal
3.2	ERP rendszerek .....	14. oldal
3.3	Adattárházak és üzleti intelligencia.....	18. oldal
3.3.1	Adattárházak rövid ismertetése .....	18. oldal
3.3.2	Üzleti intelligencia rövid ismertetése .....	20. oldal
3.4	Konklúzió .....	23. oldal
<b>4.</b>	<b>ERP rendszerek bevezetése</b> .....	24. oldal
4.1	A bevezetés lépései.....	24. oldal
4.1.1	A kiválasztás.....	24. oldal
4.1.2	Költségek és megtérülés .....	25. oldal
4.1.3	Meglevő vállalkozás .....	29. oldal
4.1.4	Induló vállalkozás.....	29. oldal
4.2	Konklúzió .....	31. oldal
<b>5.</b>	<b>ERP rendszerek testreszabhatósága</b> .....	32. oldal
5.1	Előnyök.....	33. oldal
5.2	Hátrányok .....	35. oldal
<b>6.</b>	<b>Felhasználói igények kielégítése</b> .....	37. oldal
6.1	Egy konkrét üzleti igény kielégítése.....	38. oldal
6.1.1	Oracle Applications .....	38. oldal
6.1.2	Specifikáció .....	41. oldal

6.1.3	Tervezés.....	45. oldal
6.1.4	Implementáció .....	46. oldal
6.2	Az új riport .....	52. oldal
<b>7.</b>	<b>Összefoglalás .....</b>	<b>54. oldal</b>
<b>8.</b>	<b>Köszönetnyilvánítás.....</b>	<b>56. oldal</b>
<b>9.</b>	<b>Ábrajegyzék .....</b>	<b>57. oldal</b>
<b>10.</b>	<b>Irodalomjegyzék .....</b>	<b>58. oldal</b>

# 1. Bevezetés

Az egyetemi tanulmányaim során mindig törekedtem a gyakorlatias szemlélet kialakítására, az informatikában mindig a gyakorlati hasznosságot kerestem. Az elmúlt öt évben volt szerencsém megismerkedni néhány magyarországi vállalat működésével, valamint egy külföldi közép vállalat alkalmazásában eltöltöttem több mint tizenkilenc hónapot programozóként. Ez idő alatt nagyon sok tapasztalatot szereztem a vállalatirányítás és az üzleti életben használt szoftver termékek bevezetése, felhasználása, valamint fejlesztése területén.

Az elmúlt hónapokban sok időt töltöttem az interneten vállalatirányítási rendszerekkel kapcsolatos dokumentumok, cikkek vagy tanulmányok keresésével, de nem találtam igazán jó összefoglalót vagy megfelelő dokumentációt, ami segítséget nyújtana egy kis- vagy közép vállalatnak az induláskor vagy a későbbiekben megérteni a vállalatirányítási rendszerek hasznosságát és szükségességét az üzleti életben. Az elmúlt két évben próbálkoztam konferenciákon és előadásokon bemutatni néhány hasznos információt e témával kapcsolatban, de egy előadás keretén belül nem lehetséges átfogó képet adni ilyen méretű komplex rendszerekről. Ezért most egy diplomamunka keretén belül szeretnék egy megfelelő mennyiségű összegyűjtött információval és saját tapasztalattal kiegészített tanulmányt írni az üzleti élettel kapcsolatos szoftverrendszerek és egyéb programok előnyeiről, hátrányairól valamint testreszabhatóságukról.

Reményeim szerint, ha elolvassa ezt a diplomamunkát egy kezdő fiatalokból álló csoport, akik szeretnének egy céget alapítani, vagy netán egy közép vállalati szektorban munkálkodó menedzser, aki nincs teljesen tisztában a komplex informatikai rendszerek szükségességével; akkor látni fogják, hogy a huszonegyedik században egy sikeres vállalkozás vagy vállalat életében elengedhetetlen a megfelelő informatikai háttértámogatás minden területen a megfelelő szakemberek alkalmazásával és képzésével.

Nagy hangsúlyt fogok fektetni a szoftverek testreszabhatósági előnyeinek és hátrányainak tárgyalására, mivel a legnagyobb gyakorlati tapasztalatot ezen a területen sikerült szereznem.

Természetesen a rendszerek szerkezetének bemutatása és egyéb programok adaptálhatósága sem fog kimaradni a tárgyalási részből.

Jómagam is írtam néhány hasznos programot, melyeket vállalatirányítási rendszerbe integrálva használtunk fel, így néhány alapötlettel valamint megvalósíthatósági ötlettel is fogok szolgálni. Ezekből látszani fog, hogy milyen mélységben érdemes még hozzájárulni az eredeti rendszerhez, látni fogja a kedves olvasó a testreszabás előnyeit és hátrányait egyaránt.

## 2. Fogalmak

Először is szeretnék tisztázni néhány felhasznált fogalmat. ABC szerinti sorrendbe szedtem őket, így az olvasás során bármikor visszalapozhat a kedves olvasó egy-egy idegen kifejezés vagy ismeretlen mozaik szó pontos megértése érdekében.

- **BI:** azaz Business Intelligence, magyarul üzleti intelligencia. Azon alkalmazások és technológiák összessége, amelyek lehetővé teszik egy vállalat adataihoz való egyszerű hozzáférést, ezen adatok elemzését és prezentálását. Hozzásegíti a vállalatokat ahhoz, hogy a felhalmozott információ mennyiségéből a saját folyamataikra ható tényezőket megismerjék, valamint átfogóbb ismereteket szerezzenek a hatékonyabb működés elősegítésére és ezáltal jobb üzleti döntéseket hozhassanak a vezetők.
- **CRM:** azaz Customer Relationship Management, magyarul ügyfélkapcsolat-kezelés. „Egy cég partnerei felé irányuló folyamatainak leírására vonatkozik. A CRM szoftver célja, hogy ezeket a folyamatokat támogassa, illetve hogy jelenlegi és potenciális ügyfelekkel kapcsolatos információkat tároljon. A rendszerben szereplő adatokat a cég különböző részlegei és dolgozói érhetik el.” [Wikipedia – CRM szócikk] Ennek köszönhetően lehetőség nyílik elemezni az ügyfelek adatait, szokásait és így könnyebb a vevői elégedettséget magasabb szintre emelni.
- **DSS:** azaz Decision Support System. A döntéshozók döntéshozatali képességének támogatására fejlesztett szoftver, statisztikai adatokat állít elő a masszív adatbázisból, pillanatok alatt hasznos információkhoz juthatnak az arra jogosult felhasználók. Ezen rendszerek lehetőséget adnak a megfelelő mélységig való „leásásra az adatokban”, hogy jó minőségű statisztika készülhessen és képesek egyszerűen, grafikonokon megjeleníteni az adatokat.
- **DW:** azaz Data Warehouse, magyarul adattárház. Homogénné tett adatok szervezett gyűjteménye, amely a vállalati adatokból való információ kinyerést segíti elő és döntéstámogatási célokat szolgál.

- **EDP:** azaz Electronic Data Processing, ezen rendszerek jelentették az automatizált adatfeldolgozás kezdetét. Egyszerű használat és a felhasználók által könnyen tanulható ismétlődő műveletek automatikus adatbevitel jellemző rájuk.
- **EIS:** azaz Executive Information System. Gyakorlatilag egy lényegesen továbbfejlesztett MIS rendszer kifejezetten a felső vezetés számára. A kinyert adatok, összeállított statisztikák végül itt találkoznak a felső vezetés által meghatározott üzleti célokkal, segítve a végső döntést. Így képesek a belső és külső információkat egyaránt összehasonlítani a végső célokkal.
- **ERP:** azaz Enterprise Resource Planning, közvetlen fordításban Vállalati Erőforrás Tervezés, de a mindennapi használatban jellemzően Integrált Vállalatirányítási Rendszert értenek alatta: „a vállalat környezetére, belső működésére és a vállalat – környezet tranzakcióira vonatkozó információk koordinált és folyamatos beszerzését, feldolgozását, tárolását és szolgáltatását végző személyek, tevékenységek, valamint a funkciók ellátását lehetővé tevő hardver- és szoftvereszközök összessége.”[Wikipedia - Vállalatirányítási\_információs\_rendszerek szócikk].
- **MIS:** azaz Management Information System. Vezetői információs rendszer, amely nevéből is adódóan különféle információkkal látja el a menedzsereket a döntéshozatal megkönnyítése érdekében. Tulajdonképpen egy tervezési rendszer, amely segítségével lehetőség nyílik összegyűjteni, feldolgozni, tárolni és közzétenni az adatokat. Egyszerű használhatósága miatt közkedvelt a középvezetői rétegek körében és nagymértékben javítja az információ elérhetőségét. Napjainkban bevett gyakorlat, hogy párhuzamosan használják az ERP rendszerekkel, így a MIS rendszer képes táplálkozni az ERP adatbázisából.
- **MRP I:** azaz Material Requirements Planning, magyarul anyag-igény tervezési rendszer. Ez volt az első komoly lépcsőfoka az integrált vállalatirányítási rendszerek kialakulásának. Az alábbi kérdésekre ad választ: Mit kell gyártani? Milyen anyagok szükségesek a gyártáshoz? Mi az ami rendelkezésünkre áll és mit kell beszerezni? Az MRP I egy fejlettebb változata a zárláncú MRP. Ezen rendszer már képes időbeliséget

is kezelni, azaz a rendelési határidőket és egyéb ütemezéseket képes összhangban tartani a gyártással és a beszerzéssel úgy, hogy figyelembe veszi a folyamatos visszajelzéseket és a munkafolyamatok időbeliségét.

- **MRP II:** azaz Manufacturing Resource Planning, magyarul gyártási erőforrás tervezés. Az integrált vállalatirányítási rendszerek következő nagy lépcsőfoka, az MRP I-ből vagy inkább a zártláncú MRP rendszerek továbbfejlesztéseként jöttek létre. A rendszer a termelővállalatok erőforrásigényének hatékony kezelését segíti elő. A „jövőbe látás” megvalósításához már szimulációs eszközöket használ. Egymással összekapcsolódó műveletek elvégzésére képes, azaz már megvalósítja a különböző területek közötti kommunikációt, adatáramlást. Célja, hogy a termelési, pénzügyi és marketing szempontok együtt érvényesüljenek a tervezési folyamatban.
- **OLAP:** azaz Online Analytical Processing, magyarul valós idejű adatelemzés. Az OLAP szoftverek a vállalati adatbázisok rendszerezett, konzisztens és ellenőrzött adatai felett egyfajta valós idejű adatelemzést tesznek lehetővé. Az első rendszerek a szimpla jelentéskészítésekből fejlődtek ki, amelyek még nem voltak kellőképpen gyorsak és rugalmasak a döntéshozók számára, ám manapság ezek a problémák már megszűntek.
- **OLTP:** azaz Online Transaction Processing, tranzakció vezérelt alkalmazások, tipikusan adatbeviteli és adatkinyerési tranzakciók kezelésére. A gyakorlatban az azonnali válaszadás fogalmával azonosítják, a felhasználó valós időben követheti tranzakcióját, amely csak később fog elkönyvelődni. Ezt a technológiát használják széleskörűen az iparban, beleértve a banki rendszereket, repülési társaságokat, szupermarketeket és a gyártások során használt informatikai rendszereket is.
- **SCM:** azaz Supply Chain Management, magyarul ellátási lánc menedzsment. Egy mai modern logisztikai irányzat. Az ellátási lánc folyamat magába foglalja a nyersanyag beszerzést (beszállítás vagy kitermelés), nyersanyag raktározást, termékek legyártását, valamint késztermék raktározást és kiszállítást a vevőhöz. Azonban a folyamat vége

nem itt van, hanem magába foglalja még a termékekhez kapcsolódó szervízzolgáltatásokat, valamint hulladékkezelést és újrahasznosítást is.

- **TCO:** azaz Total Cost of Ownership, magyarul a tulajdonlás teljes költsége. Segít a vásárlóknak és a cég menedzsereinek megmutatni egy termék vagy rendszer teljes költségét.
- **TPS:** azaz Transaction Processing System, az adatokat tranzakciószerűen dolgozza fel és tölti őket az adatbázisba. Az ACID (atomicity, consistency, isolation, durability) tulajdonságok megőrzéséért felelős, valamint képes kezelni a lock-okat és a deadlock felderítést. Főbb tulajdonságai: gyors válaszadás, visszavonhatóság és üzleti szabályok használatára kényszeríthető.

### **3. Vállalatirányítási rendszerekről általánosan**

#### **3.1 Információs rendszerek hőskora**

A vállalatirányítás és az informatika a kezdetekben csak üzemgazdasági kérdésekre koncentrált. A belső/vezetői számvitel egyik meghatározó feladata, hogy támogassa a vállalati erőforrások gazdaságos felhasználásának elősegítését és a hatékony feladat végrehajtás ellenőrzését. Ezért amikor az 1950-es években megjelentek a nagyvállalatoknál az első számítógépek, elsősorban az ilyen jellegű munkák követésére és automatizálására használták őket, valamint már a korai években használták ezeket a gépeket tranzakciók rögzítésére is. Több módon valósították meg: egyszerűbb változatai az elektronikus adatbevitelt és –feldolgozást szolgáló EDP rendszerek, míg fejlettebb formáját az események - azaz a tranzakciók - vezérelte TPS rendszerek képviselik. Ezek a rendszerek nagymennyiségű és részletes, de főként csak vállalaton belüli adatok megbízható gyűjtésére és tárolására, valamint egyszerű műveletek elvégzésére szolgáltak. A TPS rendszerekben a tranzakció fogalma egy logikailag összetartozó és egységként kezelt műveletsort foglal magába, például egy nyersanyag raktárra vétele vagy egy szállítói számla lekönyvelése. Ezek általában rutinszerűen ismétlődő, strukturálható, rövid, atomi, izolálható feladatok.

Az első évtizedekben csak kötegetelt adatfeldolgozásra volt lehetőség a korlátozott kapacitások és az alacsony fejlettségi szint miatt, azonban manapság már elvárás a valós idejű (real time) tranzakció kezelés, melyet az úgynevezett OLTP rendszerek valósítanak meg. A valóságban régóta tranzakció kezelő rendszereket használnak a bankautomatáknál, a jegyrendelő-rendszereknél stb. és ez az alapgondolat maradt is, de természetesen folyamatosan van fejlődés az adatfeldolgozás és legfőképpen a további nyújtott funkciók terén, valamint az adatmegjelenítésben.

Az 1970-es években, amely még mindig nagygépes korszakot jelentett a cégek életében, belső fejlesztések eredményeképpen létrejöttek az MRP I. rendszerek, vagyis anyag-igény tervezési rendszerek, majd a további fejlesztések eredményeképpen az MRP II. rendszerek, vagyis a gyártási erőforrások tervezésének rendszerei. Ezek az alkalmazások még csak a termelési és logisztikai ágazatokat szolgálták ki, nem igazán terjedtek ki a vállalatok egészére, nem

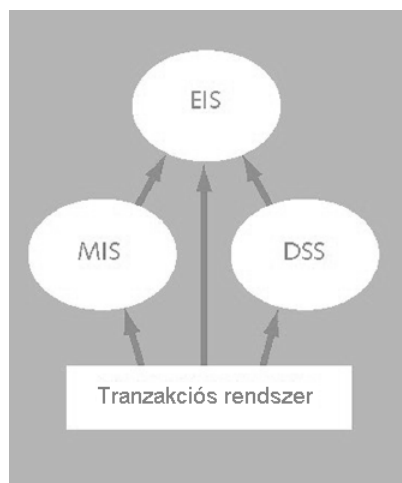
foglalkoztak a tényleges vállalati irányítással, viszont már többek voltak a pusztán tranzakciókezelő szoftvereknél.

Az 1980-as évek második felére a hatalmas informatikai fejlődéseknek köszönhetően egy magasabb szintű üzleti támogatás lehetősége fogalmazódott meg. Egyre több személyi számítógép jelent meg a különböző működési területeken és egyre nagyobb adatmennyiségeket tároltak elektronikusan. Ekkor már az üzleti vezetők is egyre gyakrabban foglalmazták meg az igényeiket, hogy ezen adattömegekből számukra hasznos kimutatások és üzleti információk jelenjenek meg közérthetően. Ez a pont volt az, ahol az informatika és az üzleti controlling együtt állt egy nagy kérdés előtt: képesek-e közösen, szoros együttműködés mellett használható információvá alakítani ezen adathalmazokat.

A vállalati vezetők feladata és felelőssége az, hogy következetes döntéseikkel előremozdítsák a céget, segítsék a fejlődést. Így elfogadhatóvá vált az igény, hogy a döntések minőségének javítása érdekében egy megfelelő információs háttérrel rendelkező személy hozza meg a kockázatos döntéseket. Így kezdtek el szépen lassan az eleinte csak automatizálási rendszerek átforni a menedzseri munkát támogató információs rendszerekké. A 80-as években létrejöttek a különböző rétegeket kiszolgáló alkalmazások:

- MIS rendszerek a középvezetők számára
- DSS rendszerek a döntés előkészítők számára
- EIS rendszerek a felső vezetés számára

Az 1. ábra jól szemlélteti ezen rendszerek egymásra épülését:



1. ábra: *Tipikus információs rendszer felépítés a 80-as évek végéről*

Ezek a kialakult információs rendszerek a kor szemléletmódjának megfelelően működtek, de mai szemmel nézve több hibát is felfedezhetünk, amelyek napjainkban olyan hibás adatkezeléshez vagy döntéstámogatáshoz vezetnének, melyek nem megengedhetőek egy jól működő informatikai rendszerben:

- Az alsó szinten, azaz a tranzakció kezelő rendszerben, túlságosan is elszeparált alkalmazások voltak, akár külön adatbázisok használatával, így nem volt lehetőség a tökéletes együttműködésre.
- Az alsó szinten a szoftverek különböző fejlesztésűek voltak, vagy esetleg teljesen más gyártóktól vásárolt szoftverek. Ezen rendszerek kommunikációját egyáltalán nem, vagy csak nagyon sok munka befektetésével tudták elérni az informatikusok.
- Az alsó szint szeparáltságából adódóan a felsőbb rétegek is szeparálttá váltak, nem tudtak a különböző területek adatokat cserélni a megbízhatóbb eredmények elérése érdekében.
- A tranzakciós rendszerek adatbázisai a feladatukból adódóan máshogy voltak optimalizálva, így nem voltak elég hatékonyak a komplex lekérdezések megválaszolásához.
- A döntéstámogatásnál alapvető probléma volt, hogy csak a belső vállalati információkat használták és sokszor csak néhány terület adataiból próbáltak válaszokat adni a menedzserek kérdéseire.

Összességében tehát a kezdeti években az információs rendszerek megvalósították a vállalaton belüli adatkezelést, valamint ezen nagymennyiségű adatból próbálták a vállalati vezetők munkáját segíteni nem túl eredményes módon.

## 3.2 ERP rendszerek

Az 1990-es évek elején további tevékenységi köröket kapcsoltak be az informatikába, így jöttek létre a mai is használatos ERP rendszerek. A lehetőségeket jól mutatja, hogy ezen rendszerek még ma is hatalmas fejlődéseken mennek keresztül. A kezdeti alkalmazások természetesen az MRP II rendszerek továbbfejlesztéseként jöttek létre, innen ered a nevük is: ERP ( Enterprise Resource Planning ), azaz vállalati erőforrás tervezés. Még manapság is a köztudatban ezzel a hárombetűs rövidítéssel utalunk ezekre a rendszerekre, azonban már sokkal többet tudnak arról, minthogy csak simán vállalati erőforrás tervező rendszereknek hívjuk őket. A napjainkban végbemenő innovációk és folyamatos bővítések eredményeképpen a helyes kifejezés az IEA ( Integrated Enterprise Application), azaz integrált vállalatirányítási alkalmazások lenne, de mint említettem a köztudatban megmaradt az ERP kifejezés. A továbbiakban, a dolgozatomban én is megmaradok a köznyelvi szokásnál, tehát az ERP szó az integrált vállalatirányítási rendszert fogja jelenteni.

Az ERP rendszerek kialakulásával az üzleti kontrolling és a menedzsment egy olyan jól használható információs bázishoz jutott, melynek segítségével és a folyamatosan bővülő lehetőségekkel egy olyan informatikai háttértámogatás lehetősége nyílt meg a cégek és vezetőik számára, amely nagyban elősegíti a cég profitorientált működését és a megfelelő döntéshozatal elősegítését.

A napjainkban használatos ERP rendszereknél továbbra is megmaradt a funkcionális szemlélet, azaz a szervezetek funkcionális és működési területeit modulokkal fedik le. Ez a moduláris felépítés könnyen összeegyeztethető a funkcionális területek sajátosságaival, azonban mégis képes az egész egy nagy szoftverrendszerre összeformálni. Nézzünk meg egy egyszerű példát:

- termelésirányítás
- értékesítés
- logisztika
- számvitel

A példán jól látszódik egy valamiféle produktumot előállító vállalat termékeinek nyomon követése funkcionális szempontból. Gyakorlatilag a termék életciklusát láthatjuk a vállalat

perspektívájából és ezeket különféle ERP modulokkal kezelhetjük. Nagyon fontos a közös adatbázis használata, segítve az egyes modulok közötti kommunikáció lehetőségét, de ezt a későbbiekben részletesen fogom tárgyalni.

Összefoglalva tehát az ERP rendszerek feladata a már ismertetett tranzakció kezelés, illetve a vezetők számára az információ láthatóvá tétele az össz-vállalati tárolt adatmennyiségből. A napjainkra kialakult ERP technológiának így sikerült ötvöznie a korábban bemutatott tranzakció kezelési módokat: EDP és TPS, valamint a MIS és DSS rendszereket.

Itt most felmerülhet a kérdés a kedves olvasóban, hogy mi a helyzet a felsővezetői réteggel. Eleinte használták felsővezetői döntéstámogatásra is ezeket a rendszereket, de hamar rájöttek, hogy az óriásira növő relációs adatbázisokkal nem tudnak mit kezdeni a legfelső szinten továbbra sem. Így alakult ki a mai modern vezetői információs rendszer legújabb generációja, amely képes együttműködni az ERP rendszerekkel. Ezen felsővezetői rendszerek bemutatása egy másik szakdolgozat terjedelmét is kimerítené, így nagyvonalakban csak annyit mondanék el róluk, hogy a gyakorlatban párhuzamosan használják őket a multinacionális cégeknél.

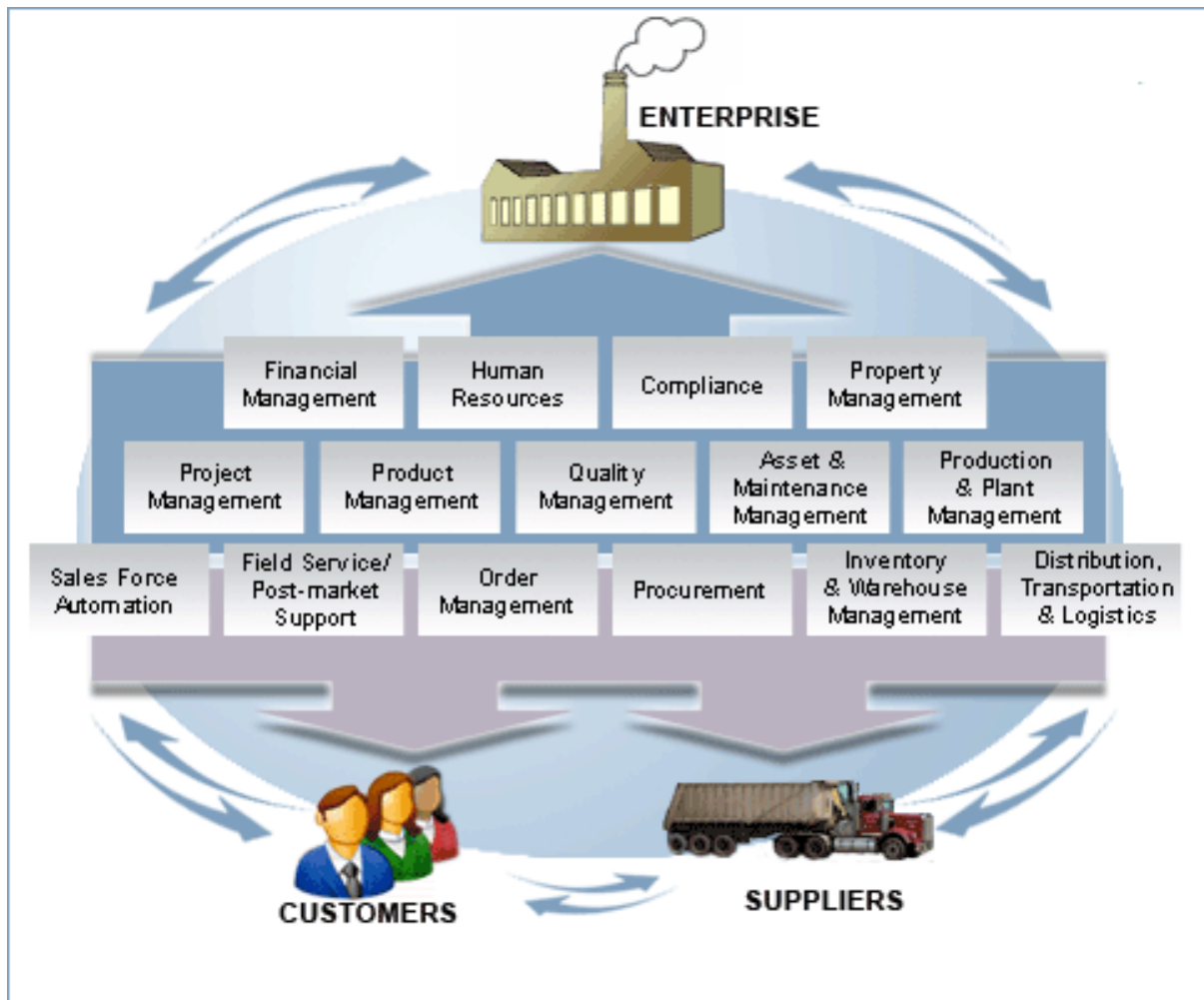
Miután kialakultak ezek a moduláris és funkcionális szemléletek, valamint körvonalazódott a szakemberek szemében az ERP rendszerek felépítésének a lehetősége, elkezdtek a nagy szoftvergyártók úgynevezett standard szoftvereket piacra dobni. Ezen szoftverek számára egy-egy általános üzleti modelltől levezetve határozták meg a területeket és hasznos, sokoldalú támogatást nyújtottak a vállalati tevékenységek informatikai automatizálására. A vállalatok is felismerték ezt és elkezdték bevezetni, majd testreszabták a szoftvereket vagy ezek egyes részeit. Valójában ezeknek megvolt az előnye és hátránya is. Egy meglévő vállalatnak a bevezetés során saját felépítését is hozzá kellett valamelyest igazítani a szoftverekhez, valamint a szoftvert is testre kellett szabni amennyire csak lehetett. Ezek a szoftver bevezetések óriási tapasztalatot nyújtottak az ERP rendszereket gyártók számára, akik a cégektől kapott információkat felhasználva még tökéletesebbé tették a rendszereket. Ezen információk feldolgozása és a szoftverrendszerekbe való beépítése nagyban hozzájárult ahhoz, hogy az elmúlt években, lassan évtizedekben, az ERP rendszerek sikeresen veszik az akadályokat a következő célkitűzésekre adott válaszaikkal:

- A nagyobb profit elérése érdekében csökkenteni kell a vállalat működési költségeit amennyire ez lehetséges.

- A vállalati tevékenységek végrehajtásához és menedzseléséhez hatékony informatikai háttértámogatást kell nyújtani.
- Javítani kell a cégek belső információ áramlását és lehetővé kell tenni a különböző területek szorosabb együttműködését.
- A jobb döntéshozatal révén magasabb minőségű és színvonalú termékeket kell előállítani, amely segít a vevők elégedettségén javítani. (Napjainkban ez a kérdéskör a legfontosabb kérdések közé lépett elő.)

A már korábban tárgyalt moduláris és funkcionális szemléletek mellett kialakult egy harmadik fő szempont is a tervezések során: ez a folyamatszemplélet. A folyamatok átlépik az egyes funkciók határait, egy egész folyamatstruktúrán viszik végig a termékeket, átölelnek több funkcionális területet is. A folyamatszemplélet kialakításához az egyes folyamatokat le kellett tudni írni, definiálni kellett őket. Hamar megjelentek a folyamatmodellező valamint a folyamatvezérlő programok. Napjainkban a folyamatszemplélet jegyében új területeken kezdtek el fejleszteni az ERP rendszerek képességeit, ezek az SCM illetve a CRM területek. Ezek segítségével terjedhetnek ki az ERP rendszerek határai a vállalat belső határain kívülre, elérve a vevőkört és így segítenek biztosítani a vevők állandó elégedettségét. Egy cég életében ugyanis egyaránt döntő jelentőségű a beszállítókkal és a vevőkkel kialakított viszony, a kölcsönös bizalom elve. Ekkor kezdtek el terjedni az adatpiacok és adattárházak, valamint az ezekre épülő sokoldalú OLAP rendszerű szoftverek fejlesztései is. Ezekkel a technológiákkal újraértelmezték a menedzseri döntések támogatását, az egyes szakterületek információigényét kiszolgálva teljesen új informatikai területet nyitottak: az üzleti intelligencia világát. Így könnyebbé vált az SCM és a CRM igények kielégítése is.

Végül álljon itt egy ábra a napjainkban használt ERP rendszerek általános felépítéséről, moduljairól:



2. ábra: Tipikus ERP rendszer struktúra napjainkból

Az ábrán jól láthatóak egy terméket előállító cég funkcionalitásait lefedő modulok. A folyamatszemplét megjelenését szemléltetik az ábrán feltüntetett nyilak, amelyek egyben visszacsatolási pontok is. Egy komplex folyamatot az anyag beszerzéstől kezdve a termék gyártásán és értékesítésén át a pénzügyi elszámolásokig egy meglehetősen összetett és bonyolult folyamat menedzser irányítja, gyakorlatilag a termék életciklusát figyelemmel kíséri. A visszacsatolási pontok jelentik a teljes kommunikációt: minden terület képes minden terület adatait elérni és hasznosítani akár a folyamat optimalizálás akár az előrejelzések elkészítése érdekében. Az ERP rendszerekre ráépülő felsővezetői döntéstámogató rendszerek immáron egy letisztult, teljes adatbázis adataihoz férhetnek hozzá, és szoros együttműködésben az ERP rendszerekkel előremozdíthatják a céget a helyes úton.

### **3.3 Adattárházak és üzleti intelligencia**

A dolgozat megkezdésekor nem állt szándékomban az adattárházakról és a napjainkban nagyon is divatos üzleti intelligenciáról beszélni, mindvégig úgy éreztem, hogy az ERP részt és a ráépülő szoftverrendszereket igen is külön kell választani egymástól. Végül rá kellett jönnöm, hogy fontos néhány sorban beszélni róluk. Az ERP rendszerekről való teljes összkép kialakulásához fontos megismernünk a napjainkban talán a legdinamikusabban fejlődő üzleti döntéstámogatási rendszereket is.

Napjainkra tehát kialakult az a nézet, hogy a folyamatokat és tranzakciókat a vállalatirányítási rendszerek vezérlik, valamint az adattárházak és üzleti intelligencia erre épülve magukba foglalják a valós idejű lekérdezéseket, analitikai eszközöket, előrejelzési és adatbányászati eszközöket. Ezek együtt, közösen alkotnak egy teljes összképet a modern vállalatirányítási informatika területén.

#### **3.3.1 Adattárházak rövid ismertetése**

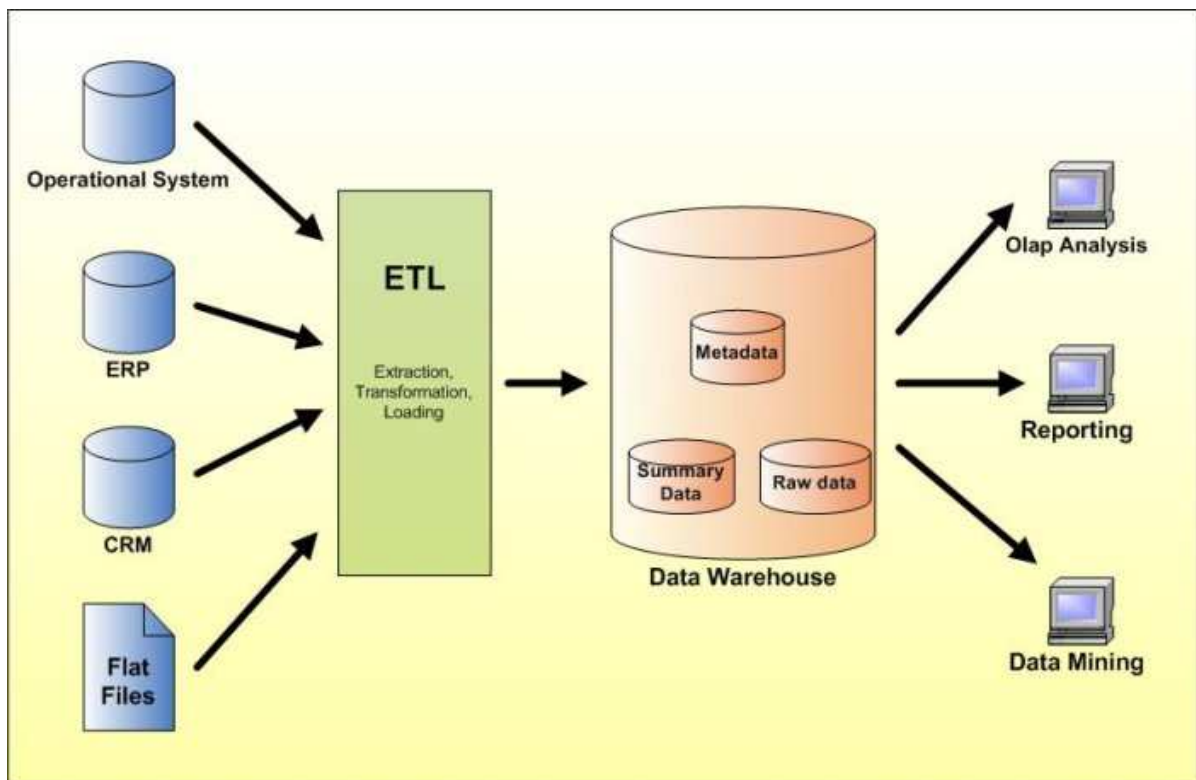
A hős korban a reportokat (jelentéseket) több napba vagy hétbe tellett megírni, így nehézkes volt bármilyen kimutatást készíteni a rendszerekből. Az OLTP rendszerek lehetővé tették a vállalatok számára, hogy ne kelljen saját alkalmazásokat írni adataik gyors kezelésére, hanem nagy és flexibilis OLTP rendszereket szabtak megfelelő formára. Az OLTP alkalmazások segítették ugyan az adatok hatékony kezelését, azonban a kihívás egy idő után a felgyülemlett adatok értelmezése lett, miként tudnak az üzlet számára értékelhető információt kinyerni. Miként lehet az óriásira duzzadt mennyiségű adatot konszolidáltan kezelni és miként lehet a vezetői és egyéb üzleti kérdésekre választ kapni ebből az adathalmazból.

Adatnak nevezhetünk egy nem értelmezett szimbólumot, jelet, egy jelentés nélküli szintaktikai fogalmat. Egy adatbázis nagyon sokféle adatot tartalmaz: megrendelési számokat, pénzmennyiségeket, életkorokat, címeket, azonosító számokat, azonban a legtöbb ezek közül önmagában nem sok információval bír. Az adattárházak, és általában véve az egész üzleti intelligencia ezekből az adatokból próbál információt kinyerni. Az információ már számunkra hasznos értelmezett adat, szemantikai fogalom. Az információt felhasználva pedig

a vállalat vezetése döntéseket hozhat. Az adattárházak tehát üzleti döntéstámogatást segítenek. Segítségükkel feltárhatjuk a trendeket egy vállalat működése mögött, használni lehet előrejelzéshez egy adott termék tervezése közben, az ügyfelek szokásainak feltérképezésére és ez a felsorolás gyakorlatilag a végtelenségig mehetne. Azokat a rendszereket, amelyek segítik ezeket a tevékenységeket, OLAP rendszereknek nevezzük.

Mit is nevezünk tehát adattárháznak? Egy olyan általános adatbázist, amely már konszolidált, integrált, aggregált, strukturált, többféle forrásból származó adatokat tartalmaz és nagyon fontos az időbeliség megjelenése, tehát időfüggő adatgyűjtemény. Ez az adatbázis ezen tulajdonságok megvalósulásával támogatja az üzleti folyamatok analizálását, és a döntéshozást.

Egy szervezeten belül több adattárház lehet. Az architektúra kialakítása során figyelembe kell venni az adatok mennyiségét, a forrásrendszerek jellegét, és egyéb más meghatározó faktorokat. Nincsen jó vagy rossz architektúra, mindig a vállalati rendszer felépítése dönti el az adattárház felépítését is.

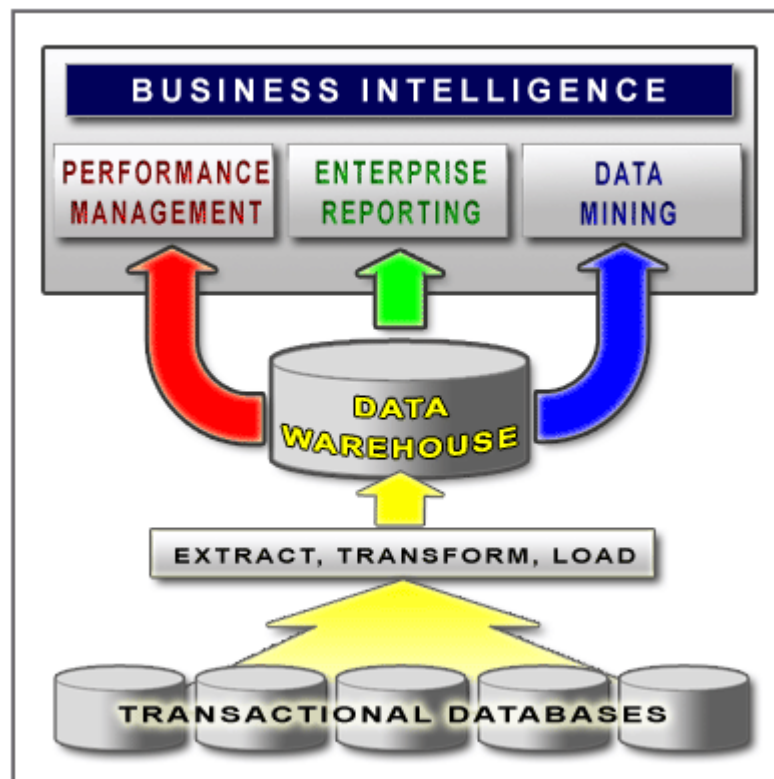


3. ábra: Egy általános adattárház architektúra

Az ábrán jól látható, hogy az adattárházba kerülés előtt egy ETL folyamaton kell végigmennie az adatoknak. A különböző forrásokból érkező adatok homogenitása nem garantált, első lépésként ki kell őket nyerni, ez az úgynevezett extract lépés. A kinyert adatok ezután egy átalakításon mennek át, közös formára kell őket hozni, ez az úgynevezett transform lépés. Végül utolsó lépésként a cél adattárházba való betöltés következik, amelyet load folyamatnak nevezünk.

### 3.3.2 Üzleti intelligencia rövid ismertetése

Az eddig ismertetett rendszerek közül a hierarchia alján találhatóak az ERP rendszerek és a hasonló integrált szoftverek, amelyek a tényleges tranzakcionális feladatokat látják el, majd ezekből dolgoznak az adattárházak és a kisebb döntéstámogató rendszerek. Végül pedig szeretném bemutatni néhány mondatban, hogy napjainkban mi az, ami a hierarchia legtetején helyezkedik el, mi az amire a multik óriási pénzeket költenek és jelenleg is rohamosan fejlődik.



4. ábra: Üzleti intelligencia a hierarchia csúcsán

Az üzleti intelligencia fogalmának meghatározásakor meglehetősen nehéz dolgunk van: meg kellene találni az üzleti intelligencia határait és egy egzakt, kézzel fogható definíciót adni. Véleményem szerint ez ma nem lehetséges, a határok a szimpla döntéstámogató rendszerek és a BI (Business Intelligence) szoftverek között elmosódnak. A legjobban talán úgy lehetne egy érthető definíciót adni, ha felsorolnánk néhány tulajdonságát:

- Segítségükkel leírhatjuk az elképzeléseinket, melyeket felhasználva a tény alapú döntéstámogatás életszerűbbé válhat.
- Segít megérteni az adatok közötti kapcsolatot, hogy a kitűzött cél felé vehessük az irányt.
- Segít a vállalat stratégiai céljait összhangba hozni az informatikai stratégiával.
- Többet nyújt, mint egy szimpla döntéstámogató rendszer.

Összefoglalva az üzleti intelligencia olyan lehetőségeket, képességeket, technológiákat valamint alkalmazásokat vagy akár gyakorlati tapasztalatra épített eljárásokat jelent, amiket az üzlet érdekében használnak és próbálják a kereskedelmi összefüggéseket feltárni. Az üzleti intelligencia alkalmazások szolgáltatnak jelen idejű, historikus, valamint megbecsült jövőbeli adatokat is az üzleti műveletekkel illetően. Az üzleti intelligencia alkalmazások rendkívül sokrétűek lehetnek ennek megfelelően, hiszen ennek a területnek az egyik részfeladata a megfelelő adatok gyors kinyerése, és annak a vezetők valamint döntéshozók felé egyszerű és letisztult prezentálása.

A kicsit nehézkes definíció megalkotása után tekintsük át az üzleti intelligencia feladatait, problémáit és azt, hogy milyen kérdésekre is kell valójában választ adnia.

A nagyvállalatok működésük során meglehetősen sok beszállítótól használnak szoftvereket: a vállalat irányítására, HR feladatokra, levelezésekre és kommunikációra, munkaidő nyilvántartásra és még sorolhatnám a különféle területeket. Ezek az alkalmazások a napi rutinszerű munka során megfelelően működnek és megfelelő felhasználói interfésszel prezentálják a saját adataikat a felhasználók felé. Azonban az üzleti intelligencia megkísérli azt, hogy mindezeket egybefogva, az adattárházakra és adatpiacokra nagyban támaszkodva, a

vállalat összességére és szokásaira nézve próbáljon egy megfelelő döntéstámogatási tervet készíteni.

Képzeljük csak el, hogy az ERP rendszerünk és a rá támaszkodó egyszerű döntéstámogató rendszerek adatait az adattárházakba kerülve egyéb vállalati és külső információkkal egybeforva tudjuk felhasználni egy magasabb szintű döntéstámogatási terv elkészítése érdekében. Mondanom sem kell, hogy jelenleg ez a meglehetősen bonyolult feladatrendszer jelenti az üzleti döntéstámogatás csúcsát.

### 3.4 Konklúzió

Az elmúlt néhány oldalban bemutatam a kezdeti éveket egy kis informatikai történelemmel fűszerezve. Beszéltem az akkori rendszerek előnyeiről és hátrányairól egyaránt. Ez reményeim szerint segít a kedves olvasónak a vállalatirányítás kialakulásának megértésében és emellett egy kis rálátást nyújt a nagy és komplex szoftverrendszerek fejlesztésének hosszú és rögzös útjaira. Az ERP rendszerek fontosságát hangsúlyoztam a következő részben, igazából ez az, ami egy kis- és középvállalat szempontjából meghatározó. Ugyanis az adattárházak és üzleti intelligencia részben többször is elhangzott az a bizonyos multi szó. Éreztettem, hogy ezek az ERP rendszerekre épülő szoftverek a nagy és multinacionális cégek világában használatosak. A kis- és középvállalatoknak természetesen nincs meg az a vezető rétege, amelyet egy szimpla ERP-ba ágyazott döntéstámogató rendszer ne tudna kiszolgálni. Ezen kis cégeknek az adatbázisai sem nőnek naponta több gigabyte adattal, nincs szükség az adattárházazásra, a meglévő adatbázisból az adatok kinyerése elfogadható sebességgel történik. Azonban a hierarchia ábrából jól látszódik, hogy a cég fejlődésével és előrelépésével az új informatikai szoftverek bevezetésének lehetősége mindig nyitva marad, mivel ezek a nagy döntéstámogató rendszerek egy az egyben ráépülnek az ERP és egyéb rendszerek adatbázisaira.

Fontos, hogy felismerje a kedves olvasó a megfelelő informatikai támogatottság szükségességét. Eleinte sok gonddal és még több ráfordítással vezethetők be ezek a rendszerek, főleg egy már meglévő vállalat számára, de hamar rá fognak jönni a kis- és középvállalati döntéshozók is, hogy hosszú távon megtérül a befektetés, sőt a cég előrelépése és a profit maximalizálása garantált lesz. Bővebben a következő fejezetben olvashat a bevezetésről és a költségekről.

## **4. ERP rendszerek bevezetése**

Az idő pénz, hangzik a közismert mondás, azonban azt is elmondhatjuk, hogy az információ pénz. A világ fejlődése, a modern technika és a versenyképesség megtartása egyre inkább rákényszeríti a kis- és középvállalatokat is, hogy integrált vállalatirányítási rendszereket vezessenek be. Ebben a fejezetben szeretném röviden ismertetni a bevezetés lépéseit, nehézségeit. Pontról pontra haladva fogja látni a kedves olvasó, hogy milyen területek milyen problémáit fogja megoldani az ERP bevezetése. E fejezet tényleges célja, hogy a kis- és középvállalatok vezetői ne riadjanak vissza e nagy volumenű változtatástól, valamint az új vállalkozást indítók már eleve egy jó üzleti stratégiával induljanak neki az üzleti élet rögzös útjainak.

A legtöbb ember azt gondolná, hogy egy vállalatirányítási rendszernek vagy csak általában véve egy informatikai rendszernek a bevezetése csupán anyagi korlátokba ütközhet. Azonban szükség van a megfelelő előkészületekre, megfelelő szakemberek alkalmazására, valamint a legfontosabb dolgok közé tartozik a dolgozók megfelelő hajlandósága, motiváltsága, munkamorálja. Így egy informatikai eszközzel a kezünkben egy megfelelő humánpolitikát kell alkalmaznunk a céljaink elérése érdekében.

### **4.1 A bevezetés lépései**

#### **4.1.1 A kiválasztás**

Amikor egy vállalat vezetése és menedzsmentje úgy dönt, hogy be kell vezetni egy vállalatirányítási rendszert, akkor elsősorban különféle döntéseket kell meghozniuk a tényleges munka megkezdése előtt:

- Célokat kell megfogalmazni, hogy pontosan mire szeretnék használni a rendszert.
- Milyen tőke áll rendelkezésre a rendszer bevezetésének teljes költségeit fedezni?
- A vállalat teljes egészét szeretnék az integrációba bevonní, vagy első lépésként csak néhány területi egységet?

- Dönteni kell a bevezetés időpontjáról, tisztán kell látni mikortól kezdve mennyi idő van az integráció véghezvitelére.

Az ezekhez hasonló kérdéseket sorolhatnám, de a legjobb, ha egy hozzáértő, tapasztalattal rendelkező tanácsadó segítségét veszik igénybe a kezdeti felmérések idején.

Az igényfelmérés után lehetséges rátérni a „Melyik rendszert válasszam?” típusú kérdésre. Nagyon sok gyártó meglehetősen sokféle vállalatirányítási rendszerrel rendelkezik, a piac széleskörű. A rendszer kiválasztásának rengeteg kritériuma van, lássunk belőle néhányat:

- Az adott rendszer képes-e lefedni a vállalat üzleti tevékenységét?
- A vállalat üzleti folyamatainak megfelelően lehet-e testre szabni a rendszert?
- Megfelelő-e a dolgozók képzettségi szintje a rendszer megismeréséhez, megtanulásához valamint használatához?
- A rendszerbe más – esetleg már használatban lévő – programokat tudunk-e integrálni?
- Milyen egyéb szolgáltatásokat lehet a rendszer bevezetésével és használatával igénybe venni?
- A tranzakciós modulokon kívül milyen egyéb szolgáltatásokra képes a rendszer? CRM, Vám, HR funkciók stb.

Tekintve, hogy a kritériumok a kitalálásukra fordított idővel egyenesen arányosan nőnek - gyakorlatilag a végtelenségig fokozhatnánk és tökéletesíthetnénk igényeinket – szükség van egy határt szabni, valamint a határon belül a kritériumokat rangsorolni. Természetesen nem fogunk olyan rendszert találni, amely teljesíti az összes elvárásunkat, ezért a cég jelenlegi helyzete és jövőbeli képe alapján kell meghozni a végleges döntést.

#### **4.1.2 Költségek és megtérülés**

A költségek alatt nem kifejezetten az anyagi költségekre fókuszálok, bár minden erre lesz visszavezethető, hiszen az üzleti élet legmeghatározóbb tényezője. Az ERP össze fogja és integrálja a szervezeti egységeket egy nagy szoftverrendszerbe. Ennek az integrált megközelítésnek óriási haszna lehet, ha a vállalat korrektül installálja a szoftverrendszert. Már

a tervezési fázis, valamint az előkészületek is hatalmas mennyiségű pénzt emésztnek fel, de ez hamarabb megtérül, mint azt bárki is gondolná.

A minőség az idő és a költség hármast a szoftverfejlesztésből is jól ismert szabályos háromszöggel írhatjuk le:



5. ábra: *Minőség-költség-idő háromszög*

A dolgok egyensúlyát szemlélteti a háromszög. Amennyiben szeretnénk valamelyik oldalt csökkenteni, úgy fognak a háromszög oldalai torzulni. Ha csökkenteni szeretnénk a bevezetési időt a minőség megtartásával, akkor lényegesen nagyobb költségekre számíthatunk, melyet láthatunk a torzult háromszög oldalainak arányán.

A megtérülést legfőképpen abban láthatjuk, hogy a szervezet életvitele gyorsabbá és tökéletesebbé fog válni. Nézzünk meg egy egyszerű példát: a vevői megrendelést. Amikor egy vevői rendelés megérkezik egy vállalathoz, tipikusan papír alapú utazgatás kezdődik egyik szervezeti egységtől a másikig. A szervezeti egységeknek megvan a saját rendszerük, többszörösen rögzíteni kell az adatokat, a kommunikáció akadályoztatott. A rendelés állapotáról közben senki nem tud pontos és kielégítő választ adni, hiszen a raktári dolgozók nem ismerik a pénzügyi osztályon dolgozók munkáját, illetve a rendelést rögzítő dolgozók sem ismerik a logisztikát. Mindemellett a különböző részlegek dolgozói nem is férnek hozzá a másik részleg informatikai állományaihoz. Az ERP segítségével a folyamatok

automatizálhatóak, és folyamatosan naprakész információkkal rendelkezhetnek mind a dolgozók mind a vezetők. A vevők gyorsabban kapják meg a terméket, valamint a legfontosabb megtérülés a befektetésünkből, hogy kevesebb lesz a hibalehetőség.

Lássuk a tényleges befektetési oldalt. A bevezetési idő meglehetősen hosszú. A bevezetési tanácsadók és bevezetéssel foglalkozó cégek 3-6 hónapot ígérnek, azonban könnyen beletelhet 8, 12 vagy akár 24 hónapba is. Ahhoz hogy egy ERP rendszer jól működjön, nem elég csak installálni a szoftvereket, hanem bizonyos üzleti módszereket és a dolgozók munkamódszereit is meg kell változtatni. Az elsődleges felmérések alatt választ kell kapnunk azokra a kérdésekre, hogy az ERP rendszer amit választottunk mennyire illik az üzleti folyamatainkhoz. Azonban ezek a felmérések hibásak lehetnek, esetleg nem vettünk figyelembe mindent, ezen hibáink a bevezetés során fognak előbukkanni és meghosszabbítják a bevezetési időt és növelik a költségeket.

Természetesen nem csak a tanácsadásért és a szoftverek installálásáért kell fizetni, hanem magát a szoftvert is meg kell vásárolni, amely esetben szintén nem beszélhetünk kis összegről. A hardverekről még nem is tettem említést. Ezen rendszerek alá meglehetősen sok és nagy szerverre van szükség. Architektúrától függően szükségesek alkalmazásszerverek valamint adatbázisszerverek és backup szerverek. Egy ERP rendszer bevezetéséről nagyon sok információt lehet olvasni az interneten, jómagam is olvastam sok cikket a költségekről és a nem várt kiadásokról. Általában a termékekre jellemző TCO mutatóval jól meghatározhatjuk a költségeket. A Meta Group cég az elmúlt években készített egy felmérést az ERP rendszerek TCO-járól, amely már tartalmazza a hardvert, szoftvert, tanácsadást és a belső alkalmazotti költségeket. 63 céget kérdeztek meg, melyek között szerepeltek kis, közép és nagyvállalatok is a teljes spektrum lefedése érdekében. Az átlag költségre 15 millió USD-t kaptak, úgy hogy a legkisebb költség 400 ezer, a legnagyobb pedig 300 millió volt. A felmérés szerint átlagban a teljes bevezetés után 8 hónappal tapasztalták az első jeleit a megtérülésnek[[http://www.indianmba.com/Faculty\\_Column/FC994/fc994.html](http://www.indianmba.com/Faculty_Column/FC994/fc994.html)].

Természetesen az összegek tájékoztató jellegűek, azonban nagyságrendileg jó támpontot adnak a kis- és középvállalati vezetőknek.

A legfontosabb megemlíteni a teljes tisztánlátás érdekében a nem várt, azaz a rejtett költségeket. Ezek közé a költségek közé kell sorolni az adatmigráció költségét, integráció és tesztelés költségét, oktatási költségeket és végül a kifejezetten az ERP rendszer bevezetése miatt alkalmazandó munkaerő költségeit. Lássuk ezeket sorjában.

Az adatmigráció feladata, hogy az évek során felhalmozott adatokat a régi rendszerekből áttöltjük az ERP rendszerbe. A meglévő céges adatokat nem lehet elfelejteni vagy kidobni jogi és üzletpolitikai szabályok miatt sem, valamint ezek az adatok nem helyettesíthetők a cég számára semmilyen módon. A különböző üzleti területek által használt különféle szoftverek adatbázisaiban felhalmozott adatok nagy mennyiségűek és változatosak lehetnek. Nem triviális feladat az adatmigráció, általában külön csoport dolgozik csak ennek a résznek a megvalósításán. A teljes összeg természetesen nem becsülhető előre konkrétan, az adatok összefésülése és homogén formára hozása meglehetősen komplex és nehézkes feladat.

A vállalatnak természetesen szüksége lehet - és szüksége is van – a vállalatirányítási rendszereken kívül más szoftverekre is. Az ERP rendszerek és más alkalmazások közötti kapcsolatot és kommunikációt meg kell teremteni, ezt nevezzük integrációnak. Az integrációnak minden ágát meglehetősen alaposan kell tesztelni a későbbi kellemetlenségek elkerülése végett. A tesztelés újabb időt és pénzt emészt fel.

Az oktatásra érdemes nagyon sok időt szánni. Szükséges minden felhasználó megfelelő oktatása, aki valaha is használni fogja vagy csak érintkezni fog az ERP rendszerrel. Lehet az új bevezetett ERP rendszer sokoldalú, üzembiztos és nyereségesség irányába mozdító lépés, ha az oktatás elmarad és a felhasználók nem fogják ismerni és szeretni az új rendszer adta lehetőségeket, akkor az ERP rendszerünk csak egy bonyolult és használhatatlan eszköz marad. A hatékony oktatás kulcsfontosságú lépése az ütemezés. Az alkalmazottaknak a napi feladatok mellett kell új, eddig nem ismert technikákkal megbarátkozniuk. Már a rendszer bevezetése előtt el kell kezdeni a szellemi felkészítést, valamint érdemes már a tesztelési fázis végefelé, azaz bőven a bevezetés utolsó fázisaiban az eszközöket a felhasználók kezébe adni. Ez nekünk informatikusoknak is sokat segít a hibák felderítésében, valamint a felhasználók játszva tanulhatják meg a rendszer használatát és itt még nem baj, ha hibáznak.

A rendszer teljes bevezetése után az éles használat alatt szükséges további informatikai fejlesztéseket megvalósítani. Általában a bevezetést felügyelő és implementáló informatikusok maradnak a cég ezen területén, és lekérdezéseket, riportokat írnak kifejezetten a cég saját ízlésének megfelelően. Az ERP rendszereknek folyamatosan informatikai támogatásra van szükségük, a hibákat el kell hárítani, mindig akad néhány új ötlet, amit meg kell valósítani.

### **4.1.3 Meglevő vállalkozás**

Talán a már meglevő és régóta jól működő vállalkozásoknak van a legnehezebb dolga egy ERP rendszer kiválasztásánál és bevezetésénél. A korábban leírtakban ismertetett kérdéskörök, taktikák, problémák valamint költségek fokozottan érvényesek egy ilyen vállalatra. Az oktatásra hatványozottan nagy figyelmet kell fordítani és még így is lehetséges, hogy meg kell válni néhány dolgozótól, akár önszántukból, akár egyéb okokból. A költségek kalkulálása ez esetben nagyobb bizonytalansági faktort tartalmazhat, hiszen a tanácsadók sem láthatnak mindent egy régóta működő cég napi munkamenetében. Az adatmigrációból fakadó problémák természetesen itt ütnek vissza. A XXI. századnak köszönhetően gyakorlatilag minden cég használ különféle szoftvereket a különböző területeken, amelyeket majd az ERP moduljai fognak felváltani. Ahol nem váltják fel, ott az integrációról kell gondoskodni.

Véleményem szerint hatalmas költségekre lehet számítani egy ekkora komplex rendszer bevezetése során, gyakorlatilag a cég életében a napi rutinokat rúgja fel az új szoftver. Ezért is ösztönzöm arra az embereket, hogy gondoljanak időben az informatikai rendszer kiépítésére. A vállalkozás indulása megfelelő időpont erre, ahogyan a következő fejezet mutatja is ezt.

### **4.1.4 Induló vállalkozás**

Egy vállalkozás megtervezésénél fontos szempont már az elején gondolni a vállalati informatikai rendszer kialakítására. A céget alapítóknak látniuk kell azt, hogy informatikai támogatás nélkül nem lesz életképes a vállalkozásuk. Gyakorlatilag az induló cégek óriási lépéselőnyben vannak a már működő cégekkel szemben.

A korábban említett kiválasztás lépései természetesen mind igazak egy új vállalkozás indításakor is. Meg kell tervezni, ki kell választani a cég számára legmegfelelőbb vállalatirányítási rendszert, még akkor is, ha nem tudjuk teljesen világosan megfogalmazni a követelményeinket az adott pillanatban. A piacon jelenlévő ERP-t fejlesztő cégek nagyon sokféle vállalatirányítási rendszert kínálnak, a választék bő. Egy induló vállalkozás számára könnyen választhatunk a profilnak megfelelő szoftverrendszert. Mindenképpen érdemes egy ERP rendszerbe fektetni, hiszen a cég mindennapjai során úgy is rá fogunk jönni, hogy szükség van minden területen az informatikai támogatásra, így aztán különböző gyártók különböző szoftvertermékeit vásároljuk meg egyesével. Ezen szoftverek között a kommunikáció egyenlő lesz a nullával, vagy nagyon nagy pénzeket kell kifizetni az IT cégek számára, hogy valamiféle működő integrációt sikerüljön fejleszteniük. Ezek után fog a cég ugyanott járni, ahol a mai működő cégek, akik szeretnének az ERP bevezetésébe belefogni, de nem tudják, hogyan is kezdjenek hozzá. Ne feledjük, hogy legalább 30 év előnyünk van ezekkel a cégekkel szemben, és ezt a tudást és tapasztalatot hasznosítani kell!

Mint említettem, egy induló vállalkozás lépéselőnyben van. A költségek és megtérülés részben ismertetett problémák és ezek költségei, valamint a rejtett költségek közül jó néhány ki fog maradni, ha időben gondolunk a jövőre. Például az adatmigráció nehézkes és rögzös útja egy az egyben kimaradhat, hiszen az indulástól kezdve az új és korszerű vállalatirányítási rendszert használja a cég.

## 4.2 Konklúzió

Ebben a fejezetben ismertettem, hogy az ERP rendszerek bevezetése nem csupán egy szoftver installáció, hanem a cég életében gyökeresen változtat meg alapvető napi munkafolyamatokat. A cégnek és a dolgozóknak egyaránt rugalmasnak kell lenniük, alkalmazkodni kell a bevezetett új vállalatirányítási rendszerhez. Néhány egyszerű példán bemutattam, hogy az ERP miben fog többet nyújtani a különálló szoftverek helyett.

A kiválasztás lépéseit röviden ismertetve fény derült néhány fontos kérdésre, mielőtt rendszert kellene választanunk. Már a kiválasztás alatt is elkezdenek a költségek jelentkezni, és nagyon hosszú megtérülési fázisra lehet számítani. A kiválasztás utáni bevezetés hosszú és rögzös útját szemléltettem néhány példával, valamint a rejtett költségek is elemzésre kerültek. Pontról pontra látható volt, hogy a rejtett költségek váratlanul és sokszor kalkulálhatatlanul jelentkeznek.

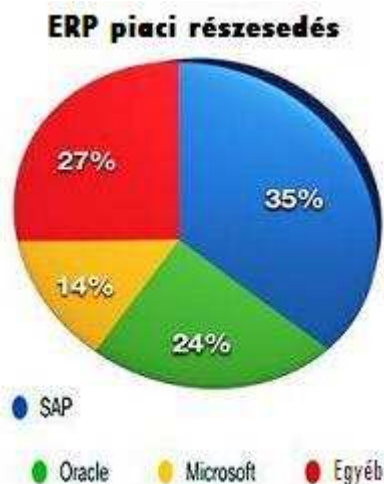
Különbséget tettem a már meglévő vállalatok és az induló vállalkozások között mind költségekben, mind lehetőségekben. Ezekben a részekben fény derült arra, hogy időben fel kell ismerni a vállalatirányítási rendszer szükségességét, és erre a legmegfelelőbb időpont a vállalkozás beindítása.

Összefoglalva elmondható, hogy meg kell ismerkedni az ERP világgal minden vállalkozást indító személynek, aki egy sikeres vállalkozás élére szeretne állni. Nem szabad visszariadni a bevezetési procedúra komplexitása vagy a költségek miatt, valamint tisztában kell lenni a rendszerek szükségességével.

## 5. ERP rendszerek testreszabhatósága

A testreszabhatóság szó alatt nem kifejezetten a rendszer adaptálhatóságát értjük, hanem inkább a saját ízlésünknek megfelelő alakítását, valamint új riportok és modulok saját fejlesztését. Az szakmai szlengben a leggyakrabban „customizálhatóságról” beszélnek, ami az angol customized szóból ered és talán ez jobban ki is fejezi a tevékenység lényegét.

Mióta az ERP rendszereket a legtöbb cég számára elérhető áron adják a fejlesztő cégek, a testreszabás kérdése többé nem olyan nagy probléma, mint évekkel ezelőtt. Maguk a fejlesztő cégek szállítanak eszközrendszert a testreszabhatóság érdekében, hiszen nekik is jó, ha látják mi az, ami az ügyfelek (jelen esetben a cégek) számára fontos fejlesztés. Ezek után mára saját új verzióik is tartalmazhatják az ésszerű fejlesztéseket. Általában a legnagyobb gyártók rendszerei a legegyszerűbb módon testreszabhatóak és a legszélesebb eszközrendszerrel rendelkeznek a customization megvalósításához. A piaci részesedések is jól mutatják, hogy a cégek a kiválasztási döntés során nagy figyelmet fordítanak erre a funkcióra is:



6. ábra: ERP piaci részesedés

Gyakorlatilag egy ERP rendszer customizálása annyit jelent, hogy egy adott terület üzleti igényei eljutnak a vállalat fejlesztő gárdájához és megpróbálják megvalósítani a felhasználók ötleteit. Más szavakkal megpróbálják feltérképezni a valóságot és ezt átültetni az ERP rendszer egy részébe, hogy jobba és könnyebbé tegye a felhasználást.

A struktúra és a folyamat megtervezése csak egy része a testreszabási folyamatnak, szükség van továbbá az input és output tesztadatok, valamint felhasználói felületek megtervezésére, validálásra, riportok és lekérdezések megírására, formátumok meghatározására, backup adatok készítésére, visszaállítási lehetőség megtervezésére és végül de nem utolsó sorban az egész folyamat adminisztrációjára. E hosszas felsorolásból látszik, hogy nem egy egyszerű rutinfeladatról beszélünk, hanem megfelelő szaktudású fejlesztő csapatra van szükség a megfelelő minőségű szoftverrendszer kialakítására.

## 5.1 Előnyök

Természetesen egy ERP szoftver vásárlása és bevezetése önmagában nem fog csodákat tenni egy vállalat mindennapjai során. Nincs olyan univerzális vállalatirányítási rendszer, amely kielégítené egy vállalat összes igényét a legkülönbébb területeken és a riportálások tekintetében. A bevezetés után az informatikai csapat nekiláthat az utómunkálatok megkezdésének. Ezek az utómunkák talán soha nem fognak befejeződni, egy nagyobb cég esetében folyamatosan szükség van az újabb fejlesztésekre és karbantartásokra. Eme fejlesztések által a vállalatirányítási rendszer jobban simul az üzlethez, gördülékenyebb a munka.

Néhány példát szeretnék bemutatni, amelyek apróságnak tűnhetnek a vállalat szemszögéből, azonban a felhasználóknak (jelen esetekben a dolgozóknak) megkönnyítik a mindennapi munkájukat:

- első példaként lássunk egy raktári visszajelzést. A raktárban tevékenykedő dolgozók a kiszállításhoz szükséges anyagokat egy úgynevezett pick lista alapján szedik össze. Ezek azok az anyagok, amik a szállító cégek dobozaiba kerülnek és mennek az adott címre. A pick listán szerepel a termék neve, raktári helye, vonalkódja és egyéb hasznos információk. A termékeket szisztematikusan kell csomagolniuk, az egybe tartozó dolgokat egy dobozba rakni. Érkezett egy kérés ezen felhasználóktól, hogy szeretnék látni a pick listán, mik az összetartozó termékek, így nem nekik kell keresgetni a számítógépben, hogy melyik termék melyikkel áll kapcsolatban. Ez egy plusz oszlopot jelent a kinyomtatandó pick lista papíron. Informatikusi szemmel

megvalósítani annyi, hogy belenyúlunk az ERP rendszer azon részébe, ami a pick listát generálja a raktárba nyomtatáskor. Fölveszünk egy új oszlopot, valamint a lekérdezést módosítjuk, ami az adatokat állítja elő a nyomtatáshoz. Gyakorlatilag nem egy túlságosan komplex feladatról van szó, nem egy projekt, csak egy kis fejlesztés. Közvetlenül az üzlet fejlődését nem segíti elő, de a felhasználók mindennapi munkáját egyszerűbbé és komfortosabbá teszi ez a fejlesztés.

- Második példaként egy termékvisszahívást elősegítő riportot szeretnék bemutatni. Kérés érkezett a cég mérnökségéről egy konkrét probléma orvoslására: az előállított terméknek az adott napon gyártott darabjai selejtesre sikerültek, ahogyan az egy utólagos teszten kiderült. Nagy munkájukba telik kézzel előkeresni, hogy melyik termék van már kint a vásárlóknál, melyik van a raktárban, illetve melyik van egy esetleges disztribúciós központban. Egy olyan riportra lenne szükségük, amely jól paraméterezzhető, és az adatokból visszaadja a termékeket a tartózkodási helyükkel és egyéb adataikkal együtt. A használatban lévő ERP rendszernek nincs ilyen funkciója, így egy teljesen új riportot kell fejleszteni, amit beleépítünk a vállalatirányítási rendszerbe. Hozzá igazítjuk az egységes felhasználói felülethez, így végeredményben úgy fog kinézni, mintha egy gyári dobozos termékről lenne szó, mintha egy funkciója lenne az ERP rendszernek.
- Harmadik példaként szintén egy cég mérnökségétől érkező kérést szeretnék bemutatni. A cég elektronikai eszközöket gyárt. Ezeket az eszközöket a gyártás közben és a gyártás végén is tesztelni kell, hogy a funkciójuknak megfelelő módon működnek-e. A teszteket úgynevezett tesztállomásokon hajtják végre. Ezeket a tesztállomásokat bizonyos számú teszt lefuttatása után kalibrálni kell, valamint karbantartási műveleteket kell végrehajtani. A tesztmérnökök kérése egy olyan új riport eszköz létrehozása, amely a paraméterezésétől függően időpontra, tesztállomásra, tesztekre valamint termékekre meg tudja mondani a tesztek számát és egyéb hasznos statisztikai adatot. A használatban lévő vállalatirányítási rendszer adatbázisát felhasználva könnyedén létrehozhatunk egy új riportot, amely illeszkedik az ERP rendszer felhasználói felületéhez és tökéletesen beilleszthető a használt modul funkciói közé. A felhasználók továbbra is csak annyit fognak látni, hogy egy újabb funkcióval bővült a rendszer menüje, azonban ez már a cég saját fejlesztései közé fog tartozni.

A példákból jól látszódik, hogy kedvünkre alakíthatjuk a rendszert egy megfelelő szaktudással rendelkező programozói gárda segítségével. Az első példa csak egy kis módosítást mutat egy már meglévő eszközön, míg a második és harmadik egy teljesen új funkció bevezetését ismerteti.

Végül, de nem utolsó sorban személy szerint nagy előnynek tartom, hogy a felhasználók érezhetik, hogy fontosak, mivel visszajelzéseik alapján javítjuk a rendszert. Talán sehol máshol nincs ilyen szoros kapcsolat a szoftveriparban a felhasználók és a fejlesztők között, mint a vállalatirányítási rendszerek testreszabása közben. Nem csak hogy azt éreztetjük a felhasználókkal, hogy fontosak, hanem sokszor többet tudnak a rendszerről mint jőmagunk, mivel naponta rutinszerűen használják, ismernek minden kikaput és egytől egyig tudják sorolni az éppen aktuális szoftververzió problémáit.

## **5.2 Hátrányok**

Természetesen, mint mindennek, a testreszabásnak is vannak hátulütői. Amennyiben lehetséges, az adott problémát a lehető legkisebb módosítással kell mindig végrehajtani, a legkevesebb gyártói kódmodosításra kell törekedni. Amennyiben egy modult túlságosan is testreszabunk, könnyen elveszíthetjük a gyártó támogatását. Ha esetleg olyan problémába ütközünk, amelyet cégen belül nem tudunk kijavítani, esetleg az üzletmenet akadozik vagy teljesen leáll emiatt, akkor a legtöbb esetben igénybe vehetjük a gyártó cég segítségét a probléma orvoslásában. Azonban ha a gyári kódot módosítottuk, azt fogják mondani az esetek 99 százalékában, hogy sajnálják, de nem tudnak segíteni, ugyanis nem az ő eredeti kódjukat használjuk, így a hiba biztosan a mi módosításunk miatt áll fent.

Másik oldala a testreszabásnak, hogy ha nincs megfelelően dokumentálva, akkor könnyen átláthatatlanná, akár teljesen kaotikussá válhat a rendszer a szakemberek számára. Bármilyen kis módosítást megfelelően kell tesztelni és dokumentálni az éles használatba küldés előtt.

Amennyibe a vállalat biztosítani szeretné a teljes gördülékenységet, nem elég csak egy programozói csapat, aki elkészíti a fejlesztéseket, a testreszabások életciklusa itt nem ér véget.

Sokkal inkább kell koncentrálni a későbbi problémamegoldásra. Fel kell állítani egy támogató csapatot, akik megfelelő mélységben ismerik a rendszert és bármikor bármilyen problémára könnyedén tudnak reagálni. Természetesen ez tovább növeli a költségeket, munkaerőt kell alkalmazni kifejezetten a vállalatirányítási rendszer karbantartására.

## 6. Felhasználói igények kielégítése

Alapvetően a vállalatirányítási rendszereket készítő cégek a modulok bevezetésével garantálják számunkra a felhasználók teljes elégedettségét, azonban ez így önmagában nem igaz, hiszen nem lehet olyan univerzális szoftvert írni, amely minden vállalat számára egyformán megfelelő lenne. A felhasználói igények kielégítése több módon is történhet, attól függően, hogy milyen és mennyire lefedett területről érkezik a kérés. Sok esetben csak egy ERP modul bevezetésével megoldhatjuk a felmerülő problémákat, azonban a gyakoribb eset az, amikor a testre szabás lehetőségéhez kell nyúlnunk. Mindig mérlegelni kell, hogy a kérések és észrevételek alapján van-e reális módja a dobozos ERP rendszer kódjának módosítása nélkül megoldani a problémát, az előző fejezetben ismertetett okokat és irányelveket szem előtt tartva.

A felhasználói igények kielégítését elsősorban hívhatjuk szoftverfejlesztésnek informatikusi szempontból, azonban inkább nevezhetjük egy hosszadalmas és bonyolult procedúrának a vállalaton belül. Ha az igényfelmérés és mérlegelési fázison túljutott egy kérés és átkerül az informatikai osztályra, akkor projekt csak abban az esetben lesz belőle, ha az informatikai szakemberek is egyetértenek a szükségességével és egy megvalósíthatósági tervet elkészítve életképesnek látják az ötletet.

Mint említettem a bevezetőben is, a legnagyobb gyakorlati tapasztalatot a testreszabás és az ERP-ba beépülő új szoftverek fejlesztésében szereztem, így a fejezet hátralévő részében egy konkrét vállalatirányítási rendszerhez fejlesztett új riportot szeretnék bemutatni, amely esettanulmányként is szolgál a fejezet lezárásaként.

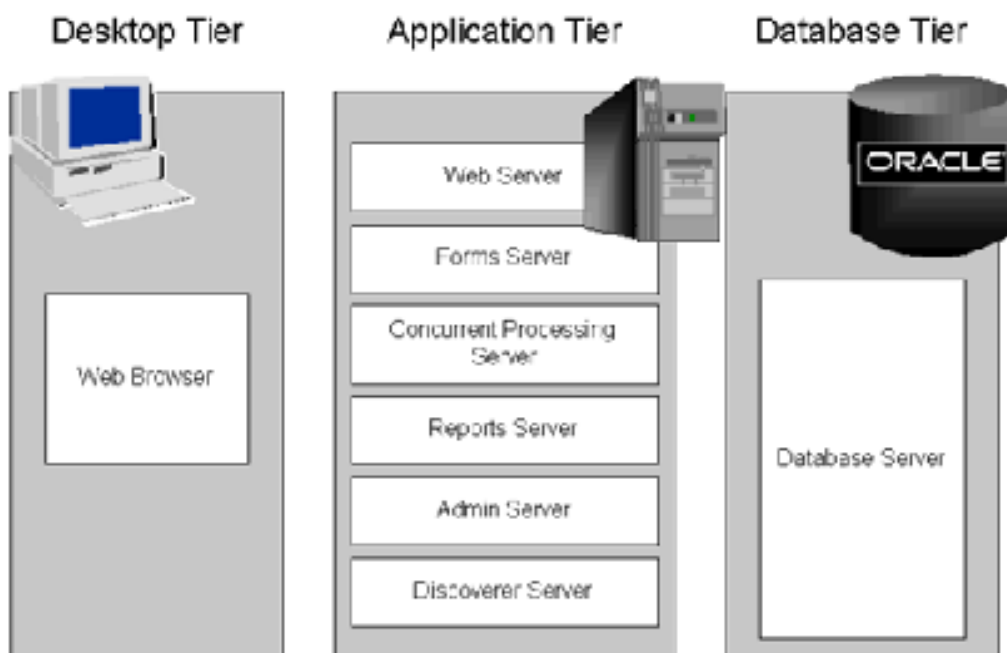
## 6.1 Egy konkrét üzleti igény kielégítése

Az esettanulmány leírásának megkezdése előtt szükségesnek tartom bemutatni néhány szóban a használatban lévő vállalatirányítási rendszert, valamint a rendszer lehetőségeit. Ezek után ismertetem a specifikációt, ami a felhasználó elmondása alapján üzleti elemzők segítségével született meg. Beszélni fogok a tervezési fázisról, hogy milyen eszközrendszerek közül volt lehetőségem választani az implementáció megkezdése előtt. Végül pedig az implementáció menetét és néhány kódrészletet fogok bemutatni az új riportból.

### 6.1.1 Oracle Applications

Az ORACLE cég által kínált vállalatirányítási szoftvercsomag. Az ERP szoftvereket használó cégek körülbelül negyede az oracle applications-t választotta, a termék testreszabhatóság és változatos, minden igényt kielégítő moduljainak köszönhetően.

Az alkalmazások lehetnek form alapúak, az ERP alkalmazások túlnyomó többsége ilyen, vagy lehetnek webes alkalmazások (JSP, PL/SQL által generált dinamikus weboldalak). A szoftvercsomag felépítése réteges, az architektúrát a következő ábra mutatja:



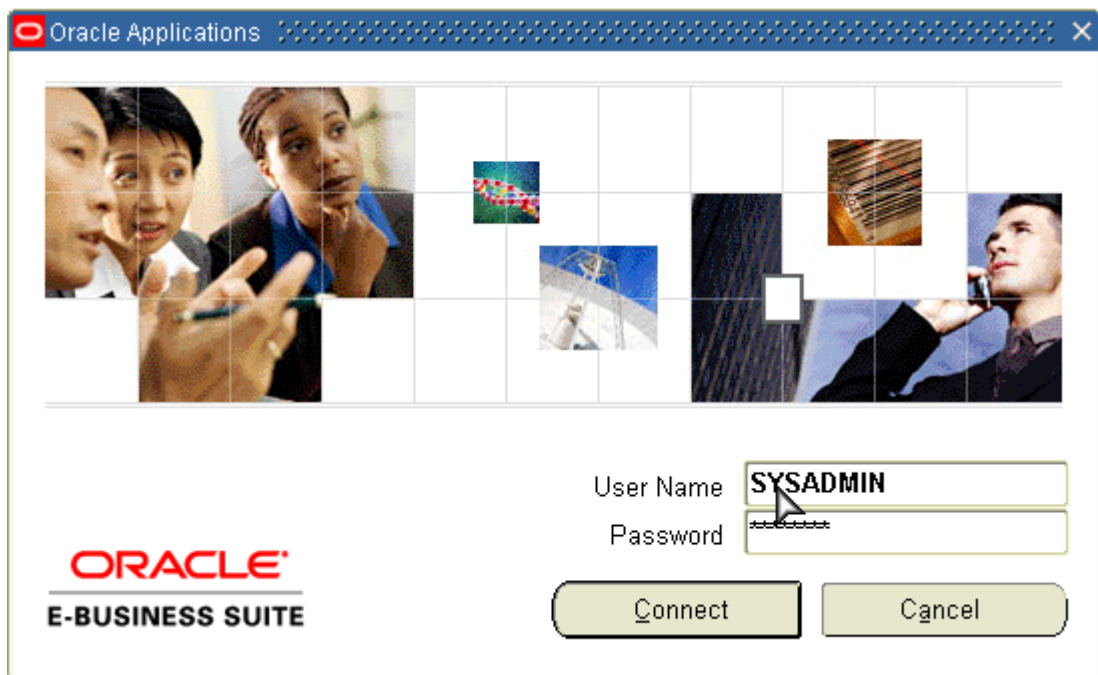
7. ábra: Oracle Applications architektúrája

Az ábra három nagy csoportosítást mutat. Az első csoport a desktop tier, ezek gyakorlatilag a kliens számítógépek, amelyek a vállalat különböző részein helyezkedhetnek el: raktár, pénzügy, termelés stb. Gyakorlatilag a felhasználók csak ezt látják az egész rendszerből, ezek a gépek kapcsolódnak a szerverekhez, a felhasználók csak ezeken keresztül érik el a rendszert. A második nagy csoport az application tier, ami a különféle alkalmazás szervereket tartalmazza, mint a form szerver, web szerver és a többi kiszolgáló. A harmadik a database tier, amely az adatbázis szerveret tartalmazza. Az architektúrán látszódik, hogy a felhasználói számítógépek csatlakoznak az alkalmazás szerverekre és ezen keresztül érik el az adatbázist.

A példában szereplő cégnél a felhasználók web böngészőt használva érhetik el az oracle applications bejelentkező képernyőjét. Az alábbi két webcím formátum elterjedt az applications elérésére:

`http://machinename:portnumber/OA_HTML/US/ICXINDEX.htm`

`http://machinename:portnumber /oa_servlets/AppsLogin`



8. ábra: *Oracle Applications* belépési képernyő

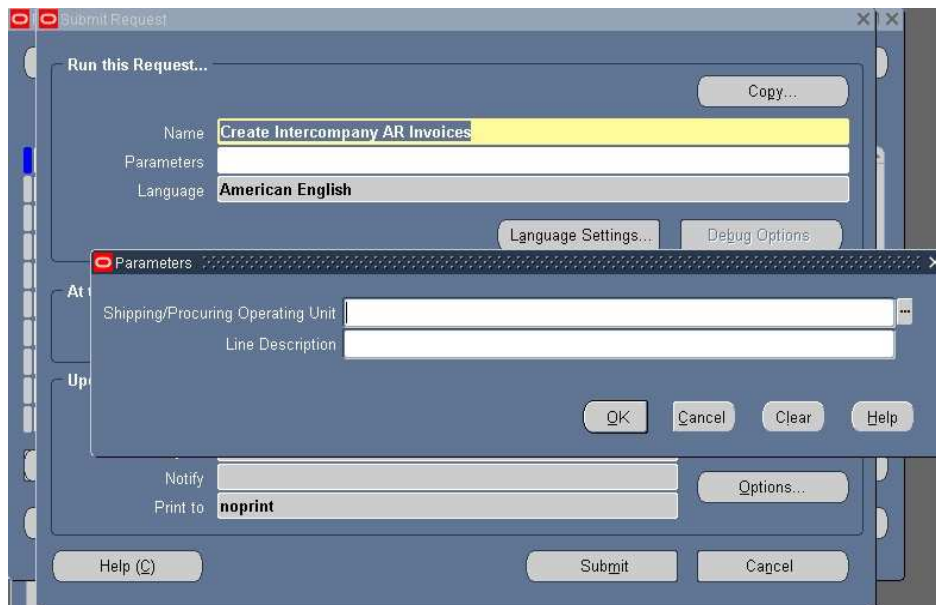
Az applications Java alapú technológiát használ a felhasználói képernyő megjelenítésére, ezért szükséges a háttérben mindig futnia a web böngészőnknek. Belépés után a használni kívánt responsibility-t választhatja ki a felhasználó, amely alapján a neki megfelelő menürendszert fogja látni:



9. ábra: Ora apps felhasználói menürendszer

A responsibility-hez tartozó navigátorbeli menüvel és a keretrendszer főmenüjével tudják elérni a kívánt funkciókat, futtathatnak úgynevezett konkurens programokat, elérhetik a modul formjait.

A következő képernyőkép szemlélteti a konkurens program futtatást, amely a főmenüből érhető el. Minden felhasználónak tudnia kell kezelni ezt a menüpontot.



10. ábra: Konkurens program futtatása

Az adott konkurens programot a felparaméterezés után a submit gomb segítségével futtathatja a felhasználó.

A néhány mondatos és képernyőképes ismertetés célja volt, hogy aki esetleg nem látta még és nem is ismeri az oracle applications vállalatirányítási rendszert, az egy rövid összefoglalóval képet kaphasson a felépítéséről és a használhatóságáról, aki pedig ismeri a rendszert, az ennél jóval mélyebben belelát akár egy mindennapi használat során szerzett tapasztalattal is.

## 6.1.2 Specifikáció

Egy termeléssel foglalkozó középállalat raktárában a dolgozók többműszakos munkarendben vannak. Vannak úgynevezett super user-ek, akik mélyebben ismerik a rendszert, ők a műszakvezetők, esetleg a raktárvezetők. Jelen esetben az alapanyagraktár kérése egy olyan riport, amivel mérni tudják a bevételezők teljesítményét: egy megadott időintervallum alatt a felhasználó mennyi bevételezési tranzakciót végez el.

### Hatáskör elemzése:

Hatásköre...	Nem hatásköre...
<ul style="list-style-type: none"> <li>• Az alapanyagraktár bevételezőinek teljesítményét méri</li> <li>• Tranzakció alapú megközelítés</li> </ul>	<ul style="list-style-type: none"> <li>• Tranzakció végrehajtásának tényleges időtartamát nem veszi figyelembe</li> <li>• Az egyéb befolyásoló tényezőket is figyelmen kívül hagyja, pl.: rendszerlassulás</li> <li>• Nem tesz különbséget az alapanyag és készáru bevételezés között.</li> </ul>

### Igénylő & Érintett terület:

Igénylők	Materials Steering Committee
Érintett területek	Alapanyagraktár

### Fogalmak:

<b>Fogalom</b>	<b>Definíció</b>
User	A felhasználó, akinek a teljesítményét méri
Nem szériaszámos tranzakció	A normál módon bevételezett, nem szériaszámos termékek száma
Szériaszámos tranzakció	A normál módon bevételezett, szériaszámos tranzakciók száma
Inspekciós tranzakció	Az inspekciósan bevételezett tranzakciók száma
Összesen	A felhasználó által végzett összes tranzakció adott időintervallumban

### Alkalmazás specifikálása :

A cégnél 3 bevételezési mód létezik: direkt, standard és inspekciós. Direkt bevételezésnél az árut egyetlen lépésben vesszük be és emeljük készletre, míg a standard bevételezés esetén ez 2 külön lépésben történik. Az inspekciós bevételezés olyan audit folyamat, ahol megvizsgálják, hogy fennállnak-e minőségi problémák az anyagoknál.

A riportban a direkt és standard bevételezéseket nem különítjük el, mert idő és munkaigényük közel azonos; ezért ezek a riportban közösen normál bevételezés névvel lesznek jelölve. Az inspekciós bevételezést azonban fontos megkülönböztetni, mivel hosszabb folyamat.

A bevételezett anyagok lehetnek szériaszámosak, melyek darabonként egyedi azonosítóval vannak ellátva, illetve nem szériaszámosak. Mivel a szériaszámos anyagokat darabonként külön kell bevételezni, ezért időigényük eltér a nem szériaszámos termékek bevételezési idejétől. Ezért a normál bevételezésen belül megkülönböztetünk normál szériaszámos és normál nem szériaszámos bevételezéseket. Az inspekciós bevételezésnél nem teszünk ilyen megkülönböztetést, mert ott mindenképp egyesével kell a termékeket megvizsgálni.

A riport az egyes bevételezők által elvégzett tranzakciók számát méri, külön bontásban a szériaszámos normál, nem szériaszámos normál és inspekciós bevételezést. A riport a következő paraméterekkel futtatható:

- Bevételező – ( felhasználó azonosító )
- Időtartam – ( dátum és óra is megadható legyen; kötelező mező )
- Organizáció – ( inventory org id; kötelező mező)
- E-mail cím

Ha a bevételezőhöz nincs név megadva, automatikusan az összes bevételezőt listázza a riport a megadott időszakban és a megadott inventory organizációra.

A riport nem ütemezve, hanem ad hoc módon lesz futtatva.

A riport állítson elő egy text file-t, ami e-mail-ben elküldhető legyen vagy a riport output állományában megtekinthető.

Az riporthoz szükséges adatokat a következő táblák tartalmazzák:

- fnd\_user tábla
  - a felhasználó neve – user\_name mező értéke, ami a user\_id alapján kereshető meg
- rcv\_transactions tábla
  - a tranzakciók száma – a riport paramétereként megadott időintervallumba belesik-e az adott tranzakció creation\_date mező értéke, és a destination\_code\_type értéke 'RECEIVING'
  - inspekciós-e vagy sem – inspection\_status\_code mező értéke NOT INSPECTED vagy INSPECTED
  - szeriaszamos vagy nem – po\_line\_id érték alapján eljutunk a po\_lines\_all táblához ( po\_line\_id ) és az ebben található item\_id mezőben levő érték alapján pedig az mtl\_system\_items\_b táblához, mégpedig az ebben a táblában található inventory\_item\_id mező értéke egyezik az po\_lines\_all táblában levő item\_id mező értékével
- mtl\_system\_items\_b (cikkszám tulajdonságai, szeriaszamos-e vagy sem).
  - Ez a tábla a po\_lines\_all táblán keresztül érhető el
  - Az inventory\_item\_id és a megadott inventory organizáció alapján kiválasztott rekordban ha a
    - serial\_number\_control\_code= 1 → nem szeriaszamos
    - serial\_number\_control\_code<>1 → szeriaszamos

- egyéb info: az inspekciós bevételezés meghatározása itt is megtörténhet az alábbi módon
  - inspection\_required\_flag = Y → inspekciós
  - inspection\_required\_flag = N → nem inspekciós

A riport szerkezete:

- Az oszlopok elválasztó karaktere: „|„
- Sorelválasztó karakter nem szükséges

User	Normál nem-szériaszámos bevételezés	Normál szériaszámos bevételezés	Inspekciós bevételezés	Összesen
User1				
User2				

A riportban megjelenítendő egyéb információk:

- futtatás időtartama
- futtató felhasználó
- futtatás időpontja
- organizáció

Ismert kockázati tényezők:

- A projekt sikerét kockáztatja, ha időközben megváltozik az üzleti igény, nem áll rendelkezésre az emberi és anyagi erőforrás.
- Amennyiben olyan üzleti igény jelentkezik, hogy legyen megkülönböztethető az alapanyag és a készáru bevételezés.

A riport tranzakció alapú, nem méri az egyes tranzakciók időigényét, illetve nem jeleníti meg azt, ha a bevételező más munkát végzett, így ez a teljesítmény nem jelenik meg a riportban. Nem mutatja az esetleges rendszerlassulásból eredő teljesítménycsökkenést.

### **6.1.3 Tervezés**

A specifikáció elkészülése valamint a felhasználókkal és üzleti elemzőkkel egyeztetés után el kellett gondolkodnom a megvalósítási lehetőségekről.

Milyen eszközrendszer áll rendelkezésemre? Szerencsére az oracle meglehetősen sok eszközrendszert biztosít a saját riport fejlesztéséhez. Lehetőségünk van használni az Oracle forms & reports fejlesztő eszközt, SQL vagy PL/SQL szkriptet is írhatunk, lehetőségünk van unix scriptet írni, meglehetősen öreg technikák használatára is ad lehetőséget: SQR riport és még számos eszköz közül válogathatunk.

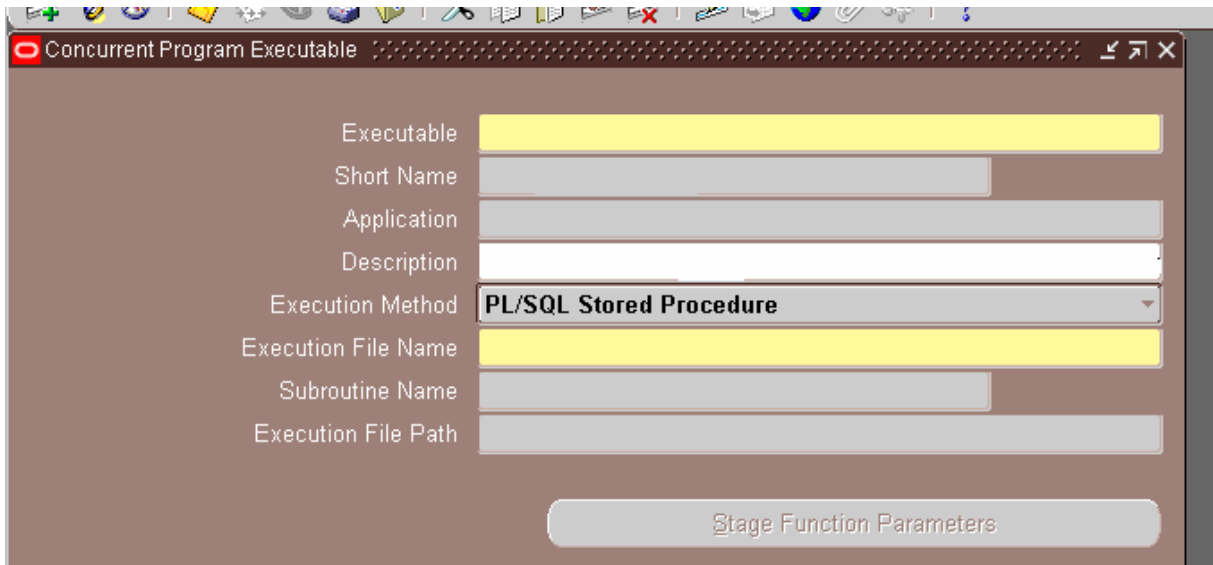
A választásom PL/SQL kód írására esett. A manapság divatos technikákat követve egy PL/SQL csomag megírása mellett döntöttem, amelyet egy konkurens program fog felhasználni. A csomag fejében lévő nyilvános metódus segítségével fog kommunikálni a konkurens program a csomag törzsében elhelyezkedő funkcionális metódusokkal. A kimeneti fájlt a PL/SQL eszközrendszerével fogom megírni, míg az e-mail küldés megvalósításához egy jól működő és régóta bevált standard oracle csomag eljárását fogom használni.

A projekt méretéből adódóan és a megfelelő részletességgel elkészített specifikáció segítségével könnyen kezelhetőnek tűnik egyetlen fejlesztő számára is, belátható időn belül tesztelésre kész verziót lehet az üzleti elemzők kezébe adni.

A specifikációban leírt adatokkal és a fejemben összeálló tervvel elegendő információval rendelkeztem az implementáció megkezdéséhez.

### **6.1.4 Implementáció**

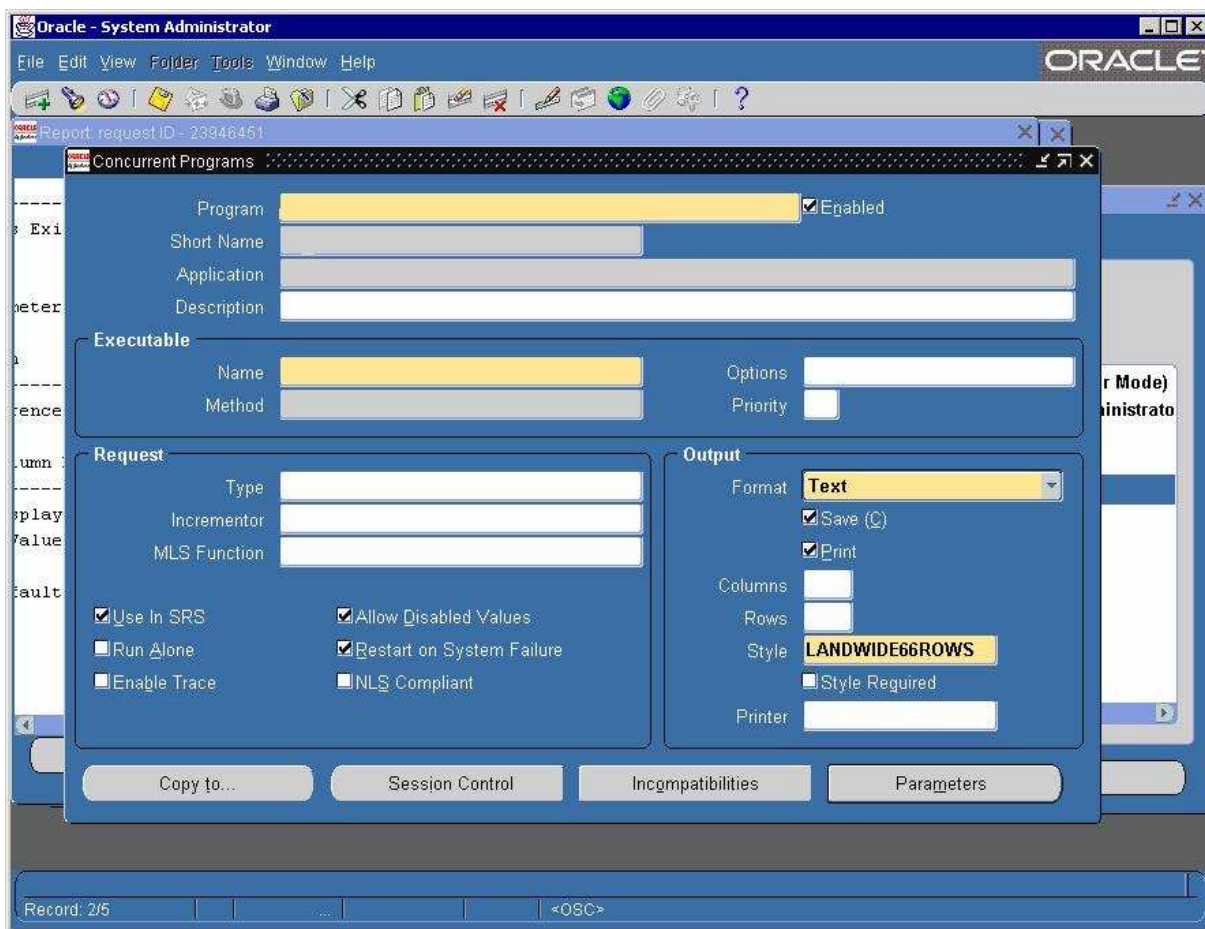
Az implementáció megkezdésekor első lépésként létrehoztam a konkurens programot, amely alá fogom írni a PL/SQL csomagot, a könnyebb programírás közbeni tesztelés érdekében. A konkurens program létrehozása előtt egy úgynevezett executable elkészítésére van szükség, amelyet majd a konkurens program fog használni.



11. ábra: Executable létrehozása

Az executable elkészítésekor definiáljuk az application-beli helyét a programunknak, valamint hogy milyen végrehajtási móddal próbáljon kommunikálni az application, azaz milyen eszközrendszerrel készítjük a kódot. Jelen esetben ez PL/SQL tárolt eljárás.

Következő lépésként létrehozhatjuk a konkurens programunkat. Az oracle applications nagyszerű eszközrendszerének köszönhetően a rendszerben egy konkurens program létrehozására szolgáló form segítségével meg is tehetjük ezt. Természetesen a form mögött rejtőző valóság az, hogy minden adatbázis szinten tárolódik, azaz a formon kitöltött adatok és a konkurens program úgyszintén az adatbázisban fog tárolódni. A név és az applicationbeli elhelyezés utáni részben meg kell adnunk a már előre általunk létrehozott executable-t. A program paraméterezését a parameters nevű gombra kattintva adhatjuk meg. Létrehozhatunk saját value set-eket a paraméter adatmezők validálására, amelyet szintén ezen a parameters formon tehetünk meg. Jelen esetben minden rendelkezésemre állt, nem volt szükség saját validálási eszközt készíteni, csak a meglévőket használnom.



12. ábra: Konkurens program létrehozása

Az incompatibilities részben megadhatjuk, hogy a programunk melyik másik programmal nem képes együtt futni. Például használhatják ugyanazt a táblát és a lockok elhelyezése miatt egy deadlock helyzet alakulhat ki, amelyet a process manager nem fog tudni kezelni, csak úgy ha kilőjük futás közben a programjainkat. Jelen esetben erre sem volt szükség, tekintve hogy csak lekérdezéseket fogok írni a PL/SQL eljárásokba, így az inkompatibilis helyzet nem alakulhat ki (Más szempontból kialakulhatna, de nem volt indokolt, a tesztelés során nem ütköztünk problémákba).

A konkurens program megtervezése után elkezdtem a PL/SQL csomag kódolását. A csomag fej részében kötelezően kell definiálni legalább egy publikus eljárást, amely segítségével kommunikálhat a konkurens programmal. Tartalmaznia kell ennek az eljárásnak 2 db kötelező paramétert, ezek az out\_errbuf és az out\_retcode, így fogja tudni az oracle applications az eljárás visszatéréséből a program sikeres futását vagy az esetleges hibákat.

```

CREATE OR REPLACE PACKAGE rcv_performance_rep AS

    PROCEDURE rpr_interface(out_errbuf   OUT VARCHAR2,
                           out_retcode  OUT NUMBER,
                           p_start_date  VARCHAR2,
                           p_end_date   VARCHAR2,
                           p_user_name   VARCHAR2,
                           p_email      VARCHAR2,
                           p_org_id      VARCHAR2);

END rcv_performance_rep;
/

```

A többi paraméter a konkurens programon látható paraméterek, amelyeket az eljárás meghívásakor automatikusan át fog adni.

A csomag törzsében első lépésként az interfész implementálásához kezdtem hozzá, így máris lehetőségem nyílt tesztelni a konkurens program futását, hogy megfelelő-e az executable és a paraméterezés. Ez az interfész rész fogja az outputba írni a riporthoz csatolandó egyéb információkat is.

```

PROCEDURE rpr_interface
(out_errbuf   OUT VARCHAR2,
 out_retcode  OUT NUMBER,
 p_start_date VARCHAR2,
 p_end_date   VARCHAR2,
 p_user_name  VARCHAR2,
 p_email      VARCHAR2,
 p_org_id     VARCHAR2)
    IS

v_output VARCHAR2(32767) := rpad('User name', 30) || '|' ||
                           rpad('Normal non-serial receives', 30) || '|' ||
                           rpad('Normal serial receives', 30) || '|' ||
                           rpad('Inspected receives', 30) || '|' ||
                           rpad('Total receives', 30);

v_start_date DATE := to_date(p_start_date, 'YYYY/MM/DD HH24:MI:SS');
v_end_date   DATE := to_date(p_end_date,   'YYYY/MM/DD HH24:MI:SS');
v_all_txn    NUMBER;
v_normal_serial_txn NUMBER;
v_all_normal_txn NUMBER;

```

```

v_user_id fnd_user.user_id%TYPE;
v_all_inspected_txn NUMBER;
v_requested_by fnd_user.user_name%TYPE;
filehandler utl_file.file_type;

```

#### CURSOR c\_user\_ids

```

IS
  SELECT DISTINCT created_by user_id
  FROM rcv_transactions
  WHERE 1 = 1
  AND transaction_date BETWEEN add_months(v_start_date, -1)
                        AND add_months(v_end_date, 1)
  AND creation_date BETWEEN v_start_date AND v_end_date

  AND organization_id = p_org_id;

```

#### CURSOR c\_user\_ids\_all\_org

```

IS
  SELECT DISTINCT created_by user_id
  FROM rcv_transactions
  WHERE 1 = 1
  /*AND transaction_date BETWEEN v_start_date*/
  AND transaction_date BETWEEN add_months(v_start_date, -1)
                        AND add_months(v_end_date, 1)
  AND creation_date BETWEEN v_start_date AND v_end_date;

```

#### BEGIN

```

filehandler := utl_file.fopen('/tmp', 'Receiving_perform_' ||
                            TO_CHAR(fnd_global.conc_request_id) ||
                            '.txt', 'w');
utl_file.PUT_LINE(filehandler, 'Receiving Performance Report');
write_to_output('Receiving Performance Report');
utl_file.PUT_LINE(filehandler, 'Report generated at: ' ||
                 TO_CHAR(sysdate(), 'YYYY/MM/DD HH24:MI:SS'));
write_to_output('Report generated at: ' ||
               TO_CHAR(sysdate(), 'YYYY/MM/DD HH24:MI:SS'));
v requested by := rpr request user;
utl_file.PUT_LINE(filehandler, 'Report generated by: ' || v_requested_by);
write_to_output('Report generated by: ' || v_requested_by);

utl_file.PUT_LINE(filehandler, 'Start date: '
                 || TO_CHAR(v_start_date, 'YYYY/MM/DD HH24:MI:SS'));
write_to_output('Start date: ' ||
               TO_CHAR(v_start_date, 'YYYY/MM/DD HH24:MI:SS'));
utl_file.PUT_LINE(filehandler, 'End date: ' ||
                 TO_CHAR(v_end_date, 'YYYY/MM/DD HH24:MI:SS'));
write_to_output('End date: ' ||
               TO_CHAR(v_end_date, 'YYYY/MM/DD HH24:MI:SS'));
IF (p_org_id IS NULL) THEN
  utl_file.PUT_LINE(filehandler, 'All organizations');
  write_to_output('All organizations');

```

```

ELSE
    utl_file.PUT_LINE(filehandler, 'Organization: ' || TO_CHAR(p_org_id));
    write_to_output('Organization: ' || TO_CHAR(p_org_id));
END IF;
utl_file.PUT_LINE(filehandler, v_output);
write_to_output(rpad('-', 30, '-') || '|' || rpad('-', 30, '-') ||
                '|' || rpad('-', 30, '-') || '|' || rpad('-', 30, '-') ||
                '|' || rpad('-', 30, '-') || '|' || rpad('-', 30, '-') ||
                '|' || rpad('-', 30, '-'));
write_to_output(v_output);
write_to_output(rpad('-', 30, '-') || '|' || rpad('-', 30, '-') ||
                '|' || rpad('-', 30, '-') || '|' || rpad('-', 30, '-') ||
                '|' || rpad('-', 30, '-') || '|' || rpad('-', 30, '-') ||
                '|' || rpad('-', 30, '-'));

```

Az output szerkezetének összeállítása, valamint a fájlba és a standard outputra írása látható az előző kódrészletben. A formátumban a pipe-ok használata elősegíti, hogy az e-mailben elküldött fájlt könnyedén beolvashatjuk bármilyen szoftverrel, amely képes az oszlopkezeléseket megvalósítani szeparátorok segítségével (pl.: MS Excel).

A kódolás során több egyszerű eljárás megírására is szükség volt, amelyek segítségével név szerint azonosíthatjuk a felhasználókat a user\_id mező alapján, az adott concurrent request-ről bővebb információkat nyerhetünk a report könnyebb azonosíthatósága érdekében, vagy az e-mail küldési funkció megvalósítását segítik elő.

A következő kódrészlet a riportból egy függvény, amely az átadott felhasználói név alapján visszaadja a felhasználói azonosítót egy standard oracle táblát felhasználva, ahol az összes vállalatirányítási rendszerbeli felhasználó adata tárolódik:

```

FUNCTION rpr_user_id
(
    p_user_name fnd_user.user_name%TYPE)
RETURN fnd_user.user_id%TYPE
IS
    v_user_id fnd_user.user_id%TYPE;
BEGIN
    SELECT user_id INTO v_user_id FROM fnd_user WHERE user_name = p_user_name;
    RETURN v_user_id;
EXCEPTION
WHEN OTHERS THEN
    write_to_log('Exception in rpr_user_id: ' || sqlerrm);
END rpr_user_id;

```

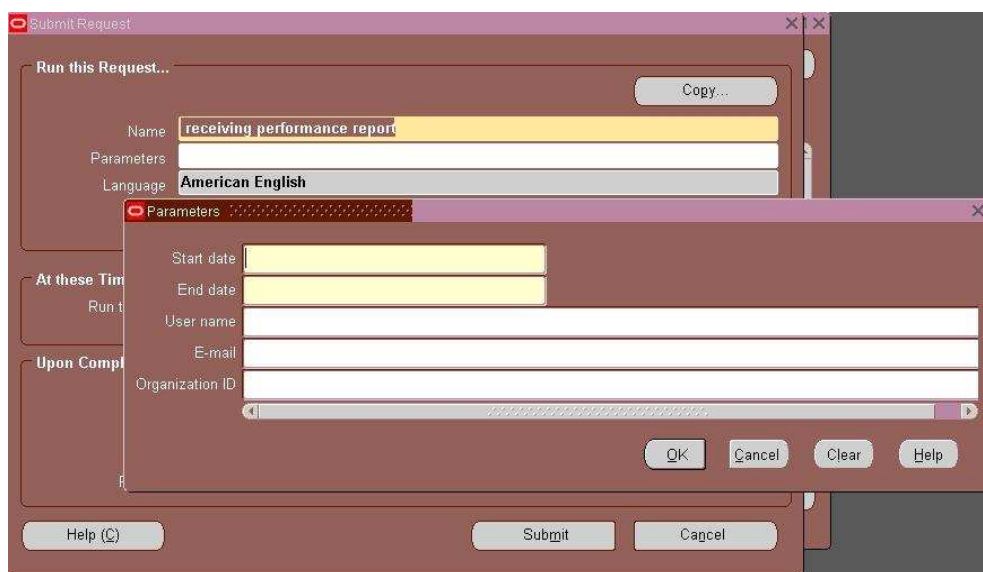
Végül azt a kódrészletet szeretném bemutatni, amellyel az összes tranzakció lekérdezését valósítottam meg:

```
FUNCTION rpr_all_transaction
(
  p_user_id rcv_transactions.created_by%TYPE,
  p_start_date DATE,
  p_end_date DATE,
  p_org_id rcv_transactions.organization_id%TYPE)
RETURN NUMBER
IS
  v_txn_number NUMBER;
BEGIN
  SELECT COUNT(*)
  INTO v_txn_number
  FROM rcv_transactions rt
  WHERE 1 = 1
  AND transaction_date BETWEEN add_months(p_start_date, -1)
                        AND add_months(p_end_date, 1)
  AND rt.organization_id = p_org_id
  AND rt.created_by      = p_user_id
  AND rt.transaction_type = 'DELIVER'
  AND rt.creation_date BETWEEN p_start_date AND p_end_date;
  IF (v_txn_number IS NULL) THEN
    RETURN 0;
  END IF;
  RETURN v_txn_number;
EXCEPTION
  WHEN OTHERS THEN
    write_to_log('Exception in rpr_all_transaction' || sqlerrm);
END rpr_all_transaction;
```

A kódban látható a transaction\_date-re való szűrés első feltételként. A transaction\_date és a creation\_date nem feltétlenül egyezik meg, viszont csak a transaction\_date oszlop tartalmazott indexet. Középvállalatról lévén szó a standard oracle rcv\_transactions tábla több milliós nagyságrendben tartalmazott sorokat. A riport futási ideje elfogadhatatlan volt a specifikációban rögzített creation\_date-re való szűrés esetén. Az üzleti elemzőkkel megbizonyosodtunk róla, hogy 1 hónapnál nem lehet nagyobb a transaction\_date mint a creation\_date semmilyen esetben, mivel havi pénzügyi zárások vannak. Így első feltételként leszűröm transaction\_date alapján a tranzakciókat egy hónapos rádiuszt figyelembe véve, majd ezen adathalmazból dolgozok tovább. Ez a módszer nagyságrendekkel lerövidítette a riport futási idejét.

## 6.2 Az új riport

A tesztelési fázis után az üzleti elemzők jónak találták a riportot, valamint a futási idő is elfogadható méretűre csökkent, így installálásra került az éles rendszerbe. Az alapanyag raktár vezetői így egy összképet láthatnak a dolgozók teljesítményéről. Végül szeretnék bemutatni néhány használat közbeni képernyőképet a riportról:



13. ábra: A riport futtatása



14. ábra: A riport futás közben

The screenshot shows a Microsoft Internet Explorer browser window. The address bar contains the URL: http://:8000/OA\_CGI/FNDWRR.exe?temp\_id=2152918932. The main content area displays a table with the following data:

User name	Normal non-serial receives	Normal serial receives	Inspected receives
HUSZELEZ	0	0	0

15. ábra: A report outputjának megtekintése web böngésző segítségével

## 7. Összefoglalás:

A diplomamunkám keretén belül általános ismeretet adtam az információs rendszerek kialakulásáról, majd a vállalatirányítási rendszerek bemutatásával foglalkoztam. Nagyon fontosnak tartom a kis- és középvállalati vezetőség szemének felnyitását, a modern informatikai támogatással ellátott vállalatirányítás megismertetését. Véleményem szerint a jelenleg elérhető szakirodalmak, cikkek, közlések e témakörben egytől egyig marketing szöveggént jelennek meg, hiszen túlnyomó többségben közgazdászok foglalkoznak a vállalatirányítással. Informatikusként egy másik perspektívából, új rálátással mutattam be a rendszereket. Úgy gondolom, diplomamunkám első harmada nagyban hozzásegíti a már meglévő vállalkozások vezetőségét a megfelelő mélységű információszerzéshez ebben a témakörben; valamint az új, induló vállalkozások vezetői is látni fogják, hogy induláskor költségeket és elvesztegetett időket spórolva vezethetik be a rendszereket. Összességében véve, véleményem szerint egy nagyon jó elméleti összefoglalót sikerült létrehoznom.

Az ERP rendszerek bevezetésének rögös útjairól úgyszintén egy megfelelő mélységű elméleti összefoglalót készítettem, amely segítséget nyújthat a bevezetési folyamatok megismerésében, úgy, hogy nem egy cég reklámja vagy a cég ügynöke mutatja be a bevezetést, hanem a nehézségekre fektetett hangsúllyal láthatóvá vált, hogy nem egy 3 hónapos projektről beszélhetünk.

A testreszabhatóságot általánosan bemutattam, segítségével megismerheti a kedves olvasó, hogy a vállalatirányítási rendszerek nem fogják teljes egészében behatárolni és irányítani a vállalat arculatát, hanem igényeinknek megfelelően alakíthatjuk a meglévő modulokat, valamint új riportokat, programokat készíthetünk, amelyeket könnyen integrálhatunk a rendszerbe, és a legtöbb esetben saját eszközrendszer áll rendelkezésünkre ezek megvalósítására.

Az üzleti igények kielégítésének lehetőségére több pontban is rávilágítottam, több példát mutatva világossá vált, hogy az ERP rendszer egy általános gyűjtő szoftverrendszere lehet a cégnek, amelyhez írhatunk saját alkalmazásokat, vagy esetleges meglévő alkalmazásokkal teremthetjük meg a kommunikációt.

Végül egy saját alkalmazásfejlesztésemet mutattam be részletesebben, amelyből jól látszik, hogy egy megfelelő szakemberekből álló gárda segítségével minimális időráfordítással elégíthető ki a cég mindennemű informatikai igénye.

A diplomamunka megírása során rá kellett jönnöm, hogy a tapasztalati előnyök felülmúlhatatlanok. Egy érdeklődő olvashat cikkeket, bemutatókat esetleg tanulmányokat, de gyakorlati tapasztalatok nélkül soha nem fogja átérezni az ERP szükségességének súlyát. Ezt szem előtt tartva igyekeztem a diplomamunka szerkezetének kialakítása során a gyakorlatiasságot hangsúlyozni, egy-egy példával vagy mindennapi rutinnal fűszerezni a mondanivalót, így reményeim szerint közelebb hoztam az olvasót a valósághoz, mint az egyéb olvasmányok.

A gyakorlati tapasztalataimat kamatoztatva szeretnék a későbbiekben is ERP rendszerek fejlesztésével, karbantartásával és testreszabásával foglalkozni.

## 8. Köszönetnyilvánítás:

Ezúton szeretnék köszönetet mondani:

- Elsősorban szüleimnek, az ő támogatásuk nélkül soha nem sikerült volna befejezni egyetemi tanulmányaimat.
- Továbbá Prof. Dr. Végh János egyetemi tanárnak, aki elvállalta témám vezetését és hasznos tanácsokkal látott el mind a diplomamunka megírása mind az egyetemi éveim mindennapjai során.
- Továbbá egy igazán nagyszerű „Global Operations” programozó csapatnak kiegészülve üzleti elemzőkkel, akik hatalmas technikai és üzleti rálátásukkal segítettek szakmai fejlődésem, és vidám perceket szereztek nekem minden nap munka közben.
- Végül de nem utolsó sorban barátaimnak, akikkel nagyon sok közös élménnyel lettünk gazdagabbak ez idő alatt is, akár egy városban töltöttük az elmúlt öt évet, akár nem.

## 9. Ábrajegyzék:

1. ábra: <i>Tipikus információs rendszer felépítés a 80-as évek végétől</i> .....	12. oldal
2. ábra: <i>Tipikus ERP rendszer struktúra napjainkból</i> .....	17. oldal ( <a href="http://www.symphonysv.com/markets/enterprise-applications.asp">http://www.symphonysv.com/markets/enterprise-applications.asp</a> )
3. ábra: <i>Egy általános adattárház architektúra</i> .....	19. oldal ( <a href="http://www.datawarehouse4u.info/">http://www.datawarehouse4u.info/</a> )
4. ábra: <i>Üzleti intelligencia a hierarchia csúcsán</i> .....	20. oldal ( <a href="http://www.avanco.com/sol_business_intel.html">http://www.avanco.com/sol_business_intel.html</a> )
5. ábra: <i>Minőség-költség-idő háromszög</i> .....	26. oldal
6. ábra: <i>ERP piaci részesedés</i> .....	32. oldal
7. ábra: <i>Oracle Applications architektúrája</i> .....	38. oldal ( <a href="http://wordpress.flexitechnologies.com/wordpress/?p=4">http://wordpress.flexitechnologies.com/wordpress/?p=4</a> )
8. ábra: <i>Oracle Applications belépési képernyő</i> .....	39. oldal
9. ábra: <i>Ora apps felhasználói menürendszer</i> .....	40. oldal
10. ábra: <i>Konkurens program futtatása</i> .....	40. oldal
11. ábra: <i>Executable létrehozása</i> .....	46. oldal
12. ábra: <i>Konkurens program létrehozása</i> .....	47. oldal
13. ábra: <i>A riport futtatása</i> .....	52. oldal
14. ábra: <i>A riport futás közben</i> .....	52. oldal
15. ábra: <i>A report outputjának megtekintése web böngésző segítségével</i> .....	53. oldal

## 10. Irodalom jegyzék:

- Wikipédia szócikkek:  
Enterprise\_resource\_planning  
Electronic\_data\_processing  
Vállalatirányítási\_információs\_rendszerek  
Transaction\_Processing\_System  
Online\_transaction\_processing  
Material\_Requirements\_Planning  
Manufacturing\_resource\_planning  
Management\_information\_system  
Supply\_chain\_management  
CRM  
Online\_analytical\_processing  
Data\_warehouse  
Business\_intelligence
- <http://www.erport.hu/index.php?id=24&L=1>
- <http://tudasmorzsak.hu/uzletiintelligencia-cikkek/46-uezleti-intelligencia/69-az-uezleti-intelligencia-kialakulasa>
- <http://www.isotanusitas.hu/hu/cikkolvas/vallalatiranyitas>
- <http://www.manta.hu/cikkek/cikk27.htm>
- <http://www.tudasmorzsak.hu/uzletvitel-cikkek/52-uzletvitel/135-mi-az-erp>
- [http://www.ebz-beratungszentrum.de/pps\\_seiten/sonstiges/erp\\_engl.htm](http://www.ebz-beratungszentrum.de/pps_seiten/sonstiges/erp_engl.htm)
- <http://www.microsoft.com/midsizebusiness/increase-business-value/customizing-erp.aspx>
- <http://www.iiitb.ac.in/ss/erp-faq/main7pg1.htm>
- <http://www.oracle.com/us/products/applications/index.htm>
- [http://www.indianmba.com/Faculty\\_Column/FC994/fc994.html](http://www.indianmba.com/Faculty_Column/FC994/fc994.html)