

Rendszerfejlesztés

BKÁE, Információrendszerek tanszék,

Molnár Bálint egyetemi docens

PhD, műszaki informatika

Budapest

2004

Tartalomjegyzék

| | | |
|-----------|---|-----------|
| 1. | BEVEZETÉS AZ INFORMÁCIÓRENDSZEREK ALAPFOGALMAIBA | 1 |
| 1.1 | MÓDSZER VAGY MÓDSZERTAN | 1 |
| 1.2 | MIT NEVEZÜNK RENDSZERNEK | 2 |
| 1.2.1 | <i>A rendszerszemléletű megközelítés.....</i> | <i>3</i> |
| 1.2.2 | <i>A rendszerszemléletű megközelítés hátrányai</i> | <i>8</i> |
| 1.2.3 | <i>A rendszerszemléletű megközelítés előnyei</i> | <i>8</i> |
| 1.2.4 | <i>Az adatközpontú megközelítés</i> | <i>8</i> |
| 1.2.5 | <i>A funkcionális megközelítés.....</i> | <i>9</i> |
| 1.3 | INFORMÁCIÓRENDSZEREK SZERVEZETEKBEN..... | 9 |
| 2. | INFORMÁCIÓRENDSZEREK FEJLESZTÉSE..... | 11 |
| 2.1 | INFORMÁCIÓRENDSZEREK MODELLEI ÉS ÁBRÁZOLÁSAI | 12 |
| 2.1.1 | <i>Modellek.....</i> | <i>12</i> |
| 2.1.2 | <i>Ábrázolások.....</i> | <i>12</i> |
| 2.1.3 | <i>A modellek stabilitásának elemzése.....</i> | <i>14</i> |
| 2.1.4 | <i>Az információrendszerek fejlesztésének főbb elvei.....</i> | <i>16</i> |
| 2.2 | MÓDSZERTANOK A GYAKORLATBAN | 17 |
| 2.3 | A FEJLESZTÉS SZAKASZAI | 23 |
| 2.3.1 | <i>Életciklus modellek.....</i> | <i>23</i> |
| 2.3.2 | <i>Információrendszer adaptációk készítésének szakaszai</i> | <i>27</i> |
| 3. | FOLYAMATOK ELEMZÉSE | 29 |
| 3.1 | BEVEZETÉS A FOLYAMATMODELLEZÉSBE | 29 |
| 3.1.1 | <i>Bevezetés az adatfolyam modellezésbe.....</i> | <i>30</i> |
| 4. | MÓDSZERTANOK..... | 35 |
| 4.1 | STRUKTURÁLT MÓDSZERTANOK | 35 |
| 4.1.1 | <i>Jobb végtermék.....</i> | <i>35</i> |
| 4.1.2 | <i>SSADM.....</i> | <i>37</i> |
| 4.2 | OBJEKTUM-ORIENTÁLT MÓDSZERTANOK | 42 |
| 5. | ÖSSZEFOGLALÁS..... | 42 |
| 6. | BIBLIOGRÁFIA | 43 |
| 6.1 | IDEGEN NYELVŰ..... | 43 |
| 6.2 | MAGYAR NYELVŰ | 45 |
| 7. | TÁRGYMUTATÓ | 46 |

Ábrák jegyzéke

| | |
|---|----|
| 1. ábra: Módszertan - Módszer - Technika - Jelölés..... | 2 |
| 2. ábra: A nyílt rendszer alkotórészei és határa | 4 |
| 3. ábra: A vezérlő rendszer részei és a funkcionális rendszer | 6 |
| 4. ábra: Információrendszer a szervezeten belül..... | 11 |
| 5. ábra: A modellek kívánatos minősége a stabilitás szempontjából..... | 14 |
| 6. ábra: Stabilitás elemzés..... | 15 |
| 7. ábra: Az információrendszerek statikus, dinamikus és eseményvezérelt nézetei.... | 16 |
| 8. ábra: A szoftver fejlesztés egyszerűsített ábrázolása..... | 17 |
| 9. ábra: A hibák eloszlása a fejlesztési ciklusban | 20 |
| 10. ábra: A felmérésekből származtatott projekt megvalósulások sikerességi százaléka | 22 |
| 11. ábra Az informatikai beruházások és az eszközökhöz viszonyított megtérülés (Strassman nyomán)..... | 23 |
| 12. ábra: A vízesés életciklus modell..... | 24 |
| 13. ábra: Szoftver életciklus V-modell | 25 |
| 14. ábra: Szoftver életciklus spirál modellje..... | 26 |
| 15. ábra: A folyamat specifikáció alkotóelemei közötti kapcsolat | 29 |
| 16. ábra: Az adatfolyam diagrammok alternatív jelölései | 31 |
| 17. ábra: Adatfolyam diagram | 32 |
| 18. ábra: Az adatfolyam ábra elemei között megengedett (adatfolyam) kapcsolatok . | 32 |
| 19. ábra: Az SSADM és a projekt ciklus | 39 |
| 20. ábra: 3-séma architektúra..... | 41 |

Definíciók jegyzéke

| | |
|---|----|
| Definíció 1-1 Rendszer | 2 |
| Definíció 1-2 Szinergia (Synergy) | 2 |
| Definíció 1-3 Részrendszer..... | 3 |
| Definíció 1-4 Információrendszer..... | 9 |
| Definíció 2-1 Stabilitás | 14 |
| Definíció 3-1 Adatfolyam modell..... | 30 |

1. Bevezetés az információrendszerek alapfogalmaiba

Az informatikában nagyon sok fogalmat használnak nem elég pontosan meghatározott tartalommal, vagy csak helyileg egységesült tartalommal, amelyet mások nem fogadnak el. Ezért ebben a jegyzetben illetve a hozzá csatlakozókban használt legfontosabb fogalmakra adunk meghatározásokat.

Ezzel az információrendszerek és fejlesztésük legfontosabb fogalmait alapozzuk meg, ami lehetővé teszi, hogy megtárgyaljuk a módszerek bizonyos technikáit.

1.1 Módszer vagy módszertan

Módszer

- a) *Speciális formájú eljárás vagy eljárások halmaza, amely bármely mentális erőfeszítést, gondolkodást igénylő tevékenységi körben előfordulhat, különös tekintettel nagy bonyolultságú feladatok megoldásánál.*
- b) *Szisztematikus (rendszeres) eljárás, technika vagy információ gyűjtési eljárás, amelyet valamely tudományág alkalmaz vagy megfelel egy tudományágnak;*
- c) *Készségeknek, tapasztalatoknak és technikáknak szervezett halmaza.*

Ebben az értelemben egy technológiai előírást, amely például informatikai rendszerek elkészítésére vonatkozik, módszernek nevezhetünk. Azonban az ilyen módszerekre a 'módszertan' elnevezés terjedt el, nemcsak magyarul.

Módszertan (Információrendszereknél)

- a) *Az ismereteknek, tudásnak olyan szervezett halmaza, amelyet a módszerekről gyűjtöttek össze. (A módszerek tanulmányozásának tudománya: egy értekezés vagy tanulmány a módszerről.)*
- b) *Egy módszer elveinek a gyűjteménye, amely minden konkrét esetben egy az adott helyzethez illesztett módszer formájában testesül meg, amely alkalmas az adott feladat megoldására.([Checkland81])*
- c) *Egy módszertan eljárások, technikák, eszközök és dokumentációs segédeszközök gyűjteménye, amely fázisokból, szakaszokból áll össze. Azonban egy módszertan több mint ezek egyszerű összege. Általában valamilyen filozófiai álláspontra helyezkedik, amely meghatározza az alapvonalait, ellenkező esetben nem lenne több mint egy recept vagy egy szakácskönyv.*

Tehát módszertan alatt egy módszer valamilyen meta-leírását fogjuk érteni, amely konkretizálható minden egyedi esetben és tartalmaz útmutatásokat a technikai megoldásokra.

Technika

Technikai (műszaki) eljárás vagy tevékenységek szervezett halmaza.

Az információrendszerek stratégia és informatikai tervezése során a terv leírásának módja nagyon fontos. Ezt a módot jelöléstechnikának, diagramtechnikának nevezzük.

Jelöléstechnika (Diagramtechnika)

Jelek, szimbólumok, karakterek (betűjelek), rövidítések rendszere, amelyek technikai (műszaki) tények, mennyiségek rögzítésére szolgál.

1.2 Mit nevezünk rendszernek

Definíció 1-1 Rendszer

Egymással kapcsolatban álló részekből áll, amelyek egy közös cél érdekében működnek együtt.

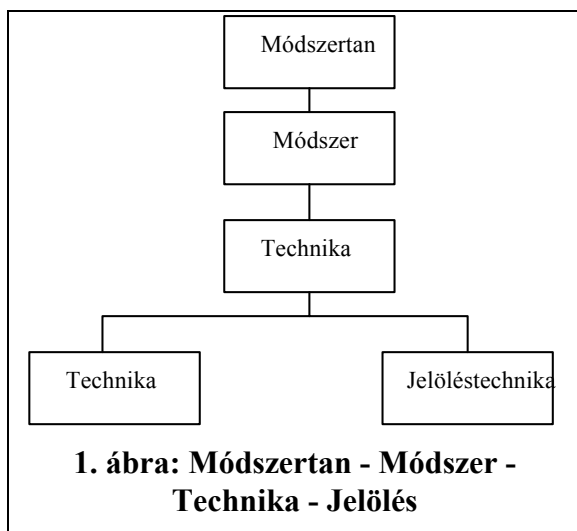
A rendszerelemzők általában egy szervezeten belül dolgoznak. Itt a 'rendszer' azt jelenti, hogy szervezett módon dolgoznak valami megvalósításáért. Kívülről nézve, a szervezeteknek van határa, amelyen túl létezik az a környezet, amelyben a rendszer (szervezet) működik. Egy szervezetet nyílt rendszernek kell tekintenünk mivel más rendszerekkel áll kapcsolatban, nevezetesen azzal a környezettel, amely a határain kívül helyezkedik el. Belülről nézve, a rendszer további alrendszereket tartalmaz,

amelyek az együttműködésen keresztül szolgálják a közös célt. Ezt az együttműködést szabályok és a külvilágból érkező impulzusok, (stimuláló) események irányítják.

Definíció 1-2 Szinergia (Synergy)

A részrendszerek hatásainak, kibocsátásaiknak az összegződése: ha közösen egy egységnek tekintjük őket, az egész nagyobb mintha csak egyedi különálló elemekként fogjuk fel őket.

Erre vonatkozik a következő idézet: "Az egész mindig nagyobb mint a részeinek összege", Arisztotelész.



A rendszerek osztályozása:

- Természetes - Ember által alkotott
- Absztrakt (pl. nyelvek, számrendszerek)
- Procedurális (pl. jogrendszer, szervezeti struktúrák)
- Fogalmi (pl. relativitás elmélet)
- Konkrét
- fizikai (pl. számítógép rendszer)
- társadalmi (pl. szervezetek)
- Zárt rendszerek
- Nyílt rendszerek (pl. információrendszer, egy szervezet, cég, stb.)

- Determinisztikus (pl. számítógép program)
- Nem-determinisztikus (pl. egy szervezetre a külvilágban bekövetkező események ill. ezek hatása)
- Valószínűségi (pl. minőség ellenőrző rendszer)
- Véletlenszerű (pl. tőzsde)
- Emberi rendszerek (pl. a szervezet stratégiai tervezői, fejlesztő csoport)
- Gépi rendszerek (pl. számítógép)
- Ember/gép rendszerek (pl. tranzakciós rendszer, vegyi üzem)
- Tervszerű rendszerek (pl. számítógépes információrendszer)
- Adaptív rendszerek (pl. szervezetek)
- Nem adaptív rendszerek
- Egyszerű rendszerek
- Bonyolult rendszerek

Definíció 1-3 Részrendszer

Részrendszer alatt önmagában megálló rendszer értünk, egyedül a kívülálló megfigyelő által elfogadott nézőpont különbözik, pl. egy szervezet információrendszere a szervezeten belül, egyszerre részrendszer és önmagában megálló rendszer.

Informatikában, emberek, gépek, módszerek halmaza, amelyek bizonyos tevékenységek végrehajtására vannak megszervezve.

1.2.1 A rendszerszemléletű megközelítés

Érdemes erről az általános megközelítésről szót ejteni. Több módszertanról állítják azt készítőik, hogy a rendszerszemléletű megközelítést követik. Az általános rendszer elméletről több alapvető munka is megjelent pl. ([Bertalanffy68],[Churchman68]). Ezek alapján érdemes egy kicsit az idetartozó fogalmakat absztraktabb módon megtárgyalni, mielőtt azokat esetleg az információrendszerek közegében használnánk és túlnyúlnak a rendszerek alap definícióján.

A rendszerszemléletű megközelítés egy egységes (holisztikus) valaminek (entitásnak, objektumnak, stb.) tekinti a rendszert nem meg feledkezve az alkotó részekről. Ez a szemlélet észleli az alkotórészek aktivitását, tevékenységét, de ugyanakkor figyelmet fordít a rendszer egésze által mutatott aktivitásra.

Néhányan ezt a megközelítést 'fekete doboz' megközelítési módnak tekintik, a rendszer bemeneteit és kimeneteit meghatározva írják le a rendszert. Természetesen, a rendszer határainak a meghatározásához, azaz a bemenetek és kimenetek pontos megadásához, meg kell mondani azokat a (rendszer)állapotokat is, amelyekhez

hozzátartoznak. Ezen a módon nemcsak egy szervezetet tekinthetünk rendszernek, hanem azt az információt is, amely a szervezeten belül található, azaz az adatokat és a hozzájuk kapcsolódó folyamatokat.

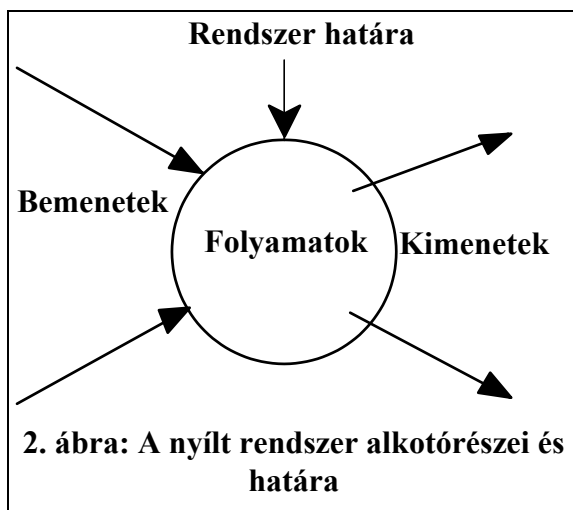
Miért érdemes a rendszerszemléletű megközelítést alkalmazni? Egyrészt a modern szervezetekben egyre gyorsuló ütemben nézünk szembe a bonyolultság és a változatosság növekedésével. A bonyolultság növekedését a következőknek tulajdoníthatjuk:

- a technológiai fejlődés;
- az értékesítési piacok kiterjedése, növekedése;
- a kutatás és fejlesztés eredményei és ennek hatása;
- az életszínvonal emelkedése;
- a termékek szakadatlan változása;
- a nemzetközi politikai és gazdasági hatásokkal az összefonódás, és a kölcsönös függőség;
- a gazdaság magán és nem-magán szférájának növekvő kölcsönös függősége.

Másrészt, új elméletek és technikák állnak rendelkezésre, amelyekkel a rendszerszemléletű megközelítést sikerrel lehet alkalmazni a vezetésben, irányításban és a szervezet információrendszereire.

Elméleti megközelítések:

- matematikai rendszerelmélet,
- kibernetikai rendszerelmélet, információ-visszacsatolás elmélete
- irányítás/vezérlés elmélet,
- információ elmélet,
- játék- és döntéselmélet,
- informatika és számítógéptudomány,



- szimuláció, stb.

1.2.1.1 A RENDSZERMODELLEK MINT ABSZTRAKT RENDSZEREK

Absztrakt rendszerek csak fogalmakból állnak - szemben a *fizikai* rendszerekkel, amelyek valamit csinálnak, valahogy viselkednek -, amelyek a fizikai valóságban nem léteznek, csak a fogalmak ideák világában. Az absztrakt rendszerek nem csinálnak semmit - ez a legfőbb megkülönböztető jel a fizikai rendszerekhez képest - azonban van

céljuk, valamilyen fizikai rendszer ábrázolása, leírása.

Egy információrendszer készítésénél általában az első lépés egy absztrakt rendszer létrehozása, amely egy szervezet bizonyos részét modellezi. A következő lépésekben ezt az induló modellt bővítik, és ily módon modellek sorozatát hozzák létre, amelyek egyre pontosabban írják le a kívánt fizikai rendszert. (Más mérnöki tudományokban is hasonló elvek szerint járnak el, pl. hídépítésnél, vagy épületek tervezésénél; az absztrakt rendszer tartalma ábrázolja a kívánt fizikai rendszert.)

1.2.1.2 A NYÍLT RENDSZEREK

Nyílt rendszer a legáltalánosabb típusa a rendszereknek, ezt mutatjuk be az ábrán (ld. 2. ábra). Vagyis egy nyílt rendszer folyamatokból áll, amelyek a környezetükből valamilyen *bemeneteket* (input) kapnak, és *kimeneteket* (output) állítanak elő. Természetesen közbenső eredmények is képződhetnek a folyamatok egyes szakaszaiban. A *rendszer határa* választja el a rendszert magát a környezetétől, és tulajdonképpen a folyamatok, a bemenetek és a kimenetek határozzák meg. A bemeneteket és kimeneteket a nyílt rendszer statikus míg a folyamatokat a rendszer dinamikus elemének tekintjük. Pl. egy cég iktatási és postázási rendszere nyílt rendszer, egy információrendszer is az.

1.2.1.3 A ZÁRT RENDSZEREK

Elméletileg, egy zárt rendszer olyan rendszer, amelyik nem lép kapcsolatba a környezetével, erre példa egy olyan kémiai 'reaktor', egy lezárt edény, amelyben valamilyen vegyi reakció folyik. Azonban, tökéletesen zárt rendszer legfeljebb laboratóriumi körülmények között létezhet csak, ha egyáltalán annak tekinthető ebben az esetben is. Sokkal hasznosabb ha zárt rendszernek azt tekintjük, amelyik a minimalizálja a környezetével zajló cserének a bizonytalansági fokát, azaz csak az előre pontosan meghatározott bemeneteket (jól-definiált inputok) fogadja el és csak előre pontosan meghatározott kimeneteket bocsát ki a feldolgozás után. (Akinek ismerősnek tűnik ez a meghatározás az nem véletlen, ez a *strukturáltság* alapelve.) Ezzel szemben, a nyílt rendszereknek meg kell birkózniuk a bizonytalan bemenetekkel, ezért nagyon alkalmazkodó képeseknek (adaptivitás) kell lenniük, pl. ilyenek az emberek, és a társadalomban létező szervezetek.

A zárt és nyílt rendszereket egy skála két szélső pontjának tekinthetjük, annak megfelelően, hogy a bemeneteiknél milyen fokú bizonytalansággal kell számolnunk. A tipikus rendszerek, amikkel a gyakorlatban találkozunk általában egyik szélsőséghez sem tartoznak, hanem köztük helyezkednek el valahol.

1.2.1.4 A RÉSZRENDSZEREK: RENDSZEREK KÜLÖNBÖZŐ SZINTEKEN

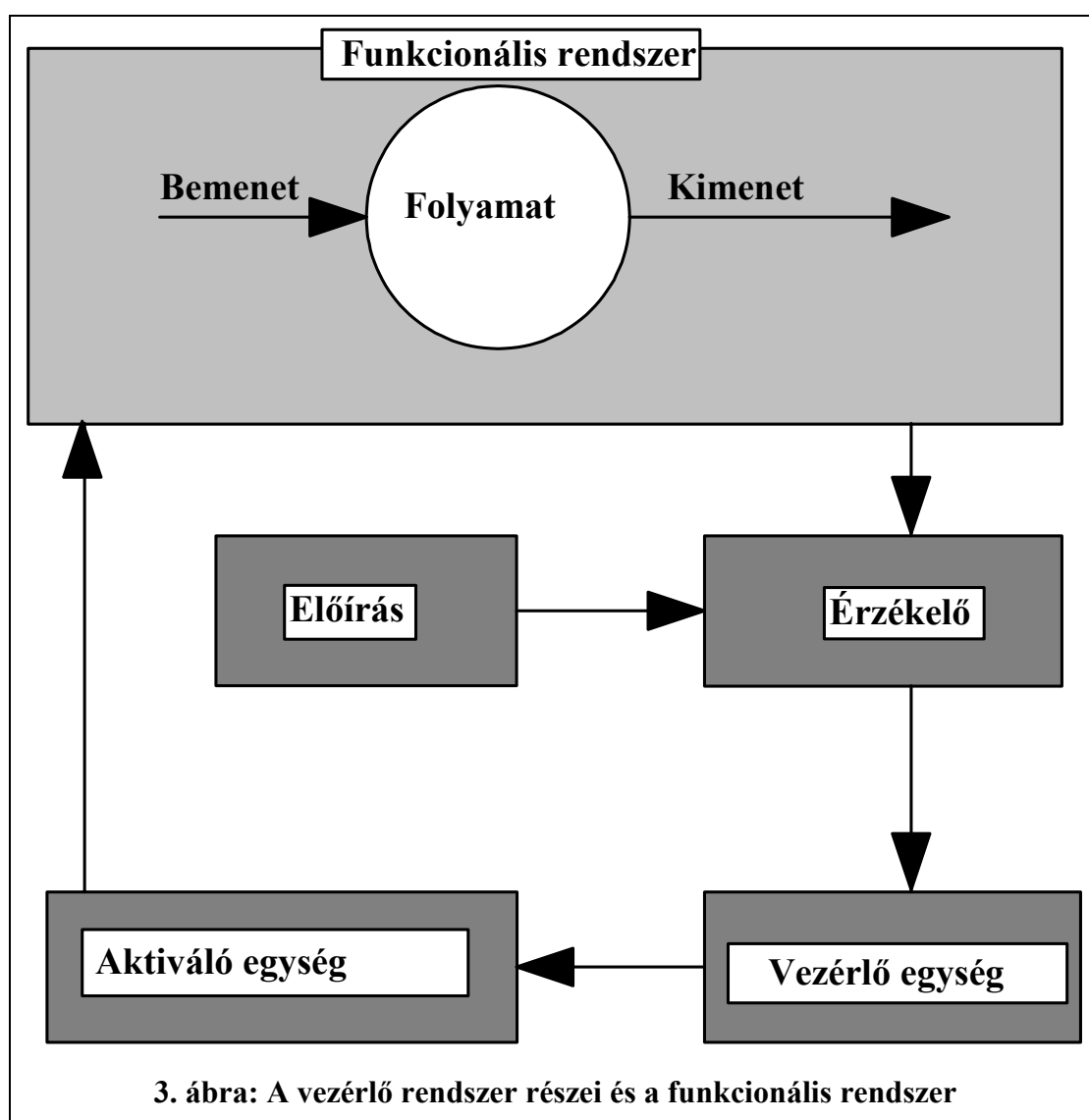
Egy rendszert leírhatunk néhány mondattal, egy diagrammal vagy akár egy több ezer oldalas dokumentummal. Egy olyan leírást, amely rengeteg részletet tartalmaz, *alacsony absztrakciós szintűnek* tekintünk, míg egy *magas absztrakciós szintű* leírás kevés részletet tartalmaz. Egy rendszert általában szisztematikus módon, részekre bontva, dekomponálva vagy finomítva (ekvivalens kifejezések) írunk le a folyamatainak magas absztrakciós szintjéből indulva és lejutva egy alacsonyabb absztrakciós szintre, amikor is a rendszer, úgy mond, *részrendszerekből* áll.

Ezt a lépés sorozatot rendszerint nagy és bonyolult rendszereknél hajtjuk végre, ezek a rendszerek olyannyira komplexek, hogy nem lehet leírni sem felfogni vagy megérteni őket egyetlen egy rendszerként. Ezért bontjuk fel olyan méretű részrendszerekre, amelyeket képesek vagyunk kezelni és megérteni. Elképzelhető, hogy még tovább bontjuk ezeket a részrendszereket is, hogy további részleteket tudjunk leírni. Gyakran valamilyen hierarchikus diagram segítségével ábrázoljuk a köztük fennálló összefüggéseket, egyes részrendszerek több másik (rész)rendszerhez tartozhatnak így egy háló formájában és nem faszerkezetben tudjuk leírni a kapcsolódásukat egymáshoz.

A már idézett definíciónk szerint (lsd. Definíció 1-3.) minden *részrendszert* önmagában rendszernek tekinthetünk és ezért további részrendszerekre bonthatjuk.

1.2.1.4.1 A kommunikáló részrendszerek

Egy nagy rendszer leírását egyszerűsíthetjük azzal, hogy részrendszerekre bontjuk, de



3. ábra: A vezérlő rendszer részei és a funkcionális rendszer

ezért komoly árat kell fizetnünk, nevezetesen meg kell határoznunk a köztük fennálló kapcsolatokat, *felületeket (interface)*, amelyeken keresztül a kommunikáció lezajlik. Egy felület alatt azokat a bemeneteket és kimeneteket értjük, amelyeket egynél több

rendszer közösen használ. Nyilvánvalóan, a bemenetek / kimenetek leírásának mindkét vagy több rendszerben létezniük kell és azonosaknak is kell lenniük. Ez különösen fontos számítógéprendszerek, - folyamatok és - programok esetén.

1.2.1.4.2 A részrendszerek csatolása

Amikor a rendszerek kölcsönhatását részletesebben vizsgáljuk, akkor időnként azt találhatjuk, hogy az egyik rendszer kimenetét egy másik rendszer azonnal felhasználja. Az ilyen eseteket nevezzük *szoros csatolásnak*. Vannak esetek, amikor ez egyáltalán nem kívánatos, mert például a fogadó rendszer csak bizonyos diszkrét állapotokban, vagy bizonyos időpontokban tudja a saját bemenetét (a másik kimenetét) feldolgozni. Egy másik ok lehet a két feldolgozási folyamat sebességének különbsége lehet.

Ekkor egy lehetséges megoldás az, ha a két rendszer szoros kapcsolatát megszüntetjük, a két rendszert *szétkapcsoljuk*, azaz a két rendszer egymástól függetlenül működik, legalábbis egy ideig. Ezt úgy valósíthatjuk meg, hogy *buffert* használunk, ahol az első rendszer kimenete várakozik addig, amíg fel nem használják.

1.2.1.4.3 Rendszerek viselkedésének kézbe tartása

Ha azt akarjuk garantálni, hogy a rendszerek céljai megvalósuljanak, folyamatosan ellenőriznünk kell a működésüket. A bemenetek lehet, hogy pontatlanok, vagy a kevésbé megértett folyamatok között összeütközések merülnek fel, ezért a rendszerhez hozzá kell kapcsolnunk egy *vezérlő rendszert*, amelyik ellenőrzi az eredeti rendszert (ezt most funkcionális rendszernek hívhatjuk), vajon helyesen viselkedik, működik-e.

1.2.1.4.3.1 Kimenet ellenőrzés

Valamilyen eszközzel észlelik a rendszer kimenetét és összevetik valamilyen előre rögzített előírással. Bármilyen eltérés egy helyreigazítási tevékenységet, korrekciót indít el, amelyet a rendszer bemenetként kap meg, ezzel működésbe hozva a funkcionális rendszert és az előírásokhoz közelebb álló kimenetet generál. A vezérlő rendszer azonban megváltoztathat folyamatokat, önszervező (adaptív) rendszereknél megváltoztathatja a rendszer célját.

Negatív visszacsatolás azt jelenti, hogyha a rendszer kimenete eltér az előírttól (valamilyen mintavételezés során), akkor a vezérlő rendszer megpróbálja az eltérést csökkenteni, a kimenetet az előíráshoz közelíteni. A pozitív visszacsatolás ennek az ellenkezőjét jelenti - ha a kimenet, annak legfontosabb jellemzői eltérnek az előírttól, akkor a rendszer megismétli az eljárást ezzel tovább növelve az előírttól az eltérést.

1.2.1.4.3.2 Bemenet ellenőrzés

A vezérlő rendszer gyakran ellenőrzi a rendszer bemenetét ugyanúgy, ahogy a kimenetét. Ezt szokták szűrésnek hívni.

1.2.1.4.3.3 A vezérlés nehézségei

A vezérlő rendszert úgy tekinthetjük mint olyant, amely csökkenti a rendszer bizonytalansági fokát. Azonban egy bonyolult rendszer vezérlése gondot okozhat, ugyanis leegyszerűsítve, minden lehetséges rendszerállapothoz tartoznia kell egy vezérlési állapotnak. Ráadásul, a rendszer elemektől vezérlési-információkat kell kapni és továbbítani nekik, ami lényegesen megnöveli a járulékos információ-feldolgozási feladatokat. Egy viszonylag nyitott vagy nem megjósolható bemenetekkel dolgozó rendszer állapotainak meghatározása gyakorlatilag lehetetlen.

Ilyen esetekben a gyakori megoldás egy ember-számítógép hibrid rendszer létrehozása, amelyben a számítógéprendszer reagál az előre megállapítható esetekre, míg az ember a nem várt esetekben hoz döntéseket.

1.2.1.4.3.4 A tervezési rendszer

Egy szervezeten belül rendszeresen felülvizsgálunk és újratervezünk rendszereket, ezért érdemes egy tervezési alrendszert is beilleszteni a funkcionális és vezérlési alrendszer mellé azért, hogy egy átfogóbb keretbe helyezzük őket. A vezérlő rendszer észleli a funkcionális rendszer kimenetének eltérését a célkitűzéstől, ezután egy a tervezési rendszer számára szóló bemenetet állít elő azért, hogy megváltoztassa a funkcionális rendszer folyamatait, bemeneteit, kimeneteit vagy célkitűzéseit. A tervezési rendszer olyan kimeneteket állít elő, amelyek az új folyamatok vagy vezérlési módok tervét tartalmazzák.

1.2.2 A rendszerszemléletű megközelítés hátrányai

1. A valós világ leírása különböző nem-egyértelmű leírása lehet az eredménye ennek a megközelítésnek; a módszer eltérő alkalmazása különböző eredményekre vezethet.
2. A létrejött modell nem teljes, vagy nem pontos, minthogy nincs pontos előírás arra, hogy milyen részletességűnek kell a leírásnak lennie.
3. Nincs kifejezetten javasolt módszer, vagy általános egyetértés abban, hogy az esetleg létező módszerek közül melyik a legmegfelelőbb.
4. " Az oroszán és a ló rendszerszemléletű megközelítésben ugyanaz, mégis egy kicsit másképpen kell velük bánni".

1.2.3 A rendszerszemléletű megközelítés előnyei

A rendszer fogalmának fentebbi meghatározásainak megfelelően közelítjük meg a valós világot vagy annak egy részét.

1. Így ez a megközelítés egy informális eszközt nyújt a helyzet megértéséhez és leírásához, az ösztönösen ismert fogalmak segítségével, mint például a folyamat, bemenet, kimenet.
2. A folyamatok *hierarchikus lebontása (dekompozíciója)* a bonyolult, összetett folyamatok elemzésének hasznos eszköze.
3. A vezérlő rendszer fogalma illetve ennek fontossága rámutathat arra, hogy egy ilyen hiányzik, nem hatékony, esetleg eredménytelen.
4. A rendszer ilyen jellegű leírása általában elég egyszerű ahhoz, hogy kommunikációs eszközként használjuk a személyek közötti párbeszéd lebonyolításához, természetesen megfelelő magyarázat kíséretében.

1.2.4 Az adatközpontú megközelítés

Ebben a megközelítésben elsősorban és hangsúlyosan azokkal az adatokkal foglalkoznak, melyek a rendszeren belül és a rendszerrel kapcsolatban felbukkannak.

Tehát meghatározzák az adattípusokat és attribútumokat majd egy adatmodellt állítanak elő, amely leírja a az adattípusok közötti kapcsolatokat. Ha helyesen és a leendő felhasználók aktív részt vételével hajtják végre, akkor azokat a dolgokat pontosan sikerült meghatározni, amelyekről a szervezet nyilvántartást akar vezetni. A modellt általában normalizálják a relációs elmélet szabályainak megfelelően, racionalizálják és végül a megvalósításhoz szükséges adatbázis tervét alakították ki. A folyamatok meghatározása ezután következik - az adattípusok egyes példányainak aktualizálása, törlése a modellnek megfelelően. Ez a megközelítés csökkenti az adatok redundanciáját, a kétértelműségeket, ön-ellentmondásokat (inkonzisztencia).

Egyes szerzők szerint ha az adatok integrációja (harmonizálása és összhangba hozása) történik meg először, akkor a szervezeti folyamatokra és tevékenységekre ez egy szükségtelen korlátozást jelent, előkészítés nélküli kényszer korlátokat jelent a folyamatokra, illetve ezek összehangolására. Más szerzők ezt kifejezetten előnyösnek tartják, mivel az adatszerkezet időben, stabil lassan változó, módosítása nagy költségeket igényel szemben a folyamatokéval, ezért az 'egészséges' adatmodellnek kell az integráció és harmonizációs munka központjában állnia.

1.2.5 A funkcionális megközelítés

A rendszert ebben a megközelítésben a funkciók hierarchiájának fogják fel. Funkciók alatt bizonyos bemenetek bizonyos kimenetekké történő *transzformációját*, átalakítását értik. A rendszer teljes egészét mit egy funkciót fogják fel és lépésenként bontják, finomítják részfunkciókká. Természetesen ebben a folyamatban a funkciók be- és kimeneteit valamint a rész- vagy alfunkciókat meghatározzák. A részfunkciók definíciói ekkor tisztázódnak, pontosan és egyértelműen definiálják őket.

A rendszerszemléletű és a funkcionális megközelítés között az alapvető különbség az, hogy a rendszerszemléletű megközelítés először a rendszer határait definiálja a bemenetek és kimenetek értelmében, míg a funkcionálisban először a funkciókat és a funkciók meghatározása során / után a be- és kimeneteket határozza meg; valójában két teljesen eltérő megközelítés.

1.3 Információrendszerek szervezetekben

Információ a kommunikáció bármely formája lehet, akár formális akár informális, amely egyben egy rendszer koherens viselkedése megértésének kulcsa.

Definíció 1-4 Információrendszer

Információrendszer a szervezet olyan része, mely információt szolgáltat, létrehoz, tárol, szétválogat, használ és eloszt. Emberi, műszaki és pénzügyi / gazdasági alkotórészekből, erőforrásokból áll. Tulajdonképpen eredendően egy humán rendszer (szervezet, manuális rendszer), mely esetleg tartalmaz egy számítógéprendszert, és ez az információrendszer bizonyos jól meghatározott részeit, kiválasztott elemeit automatizálja. Célja, hogy egy szervezet vezetési / irányítási funkcióit, valamint a mindennapi működését egyaránt támogassák

Információrendszert a következők jellemzik:

- bemenetek

Bevezetés az információrendszerek alapfogalmaiba

- kimenetek
- feldolgozási folyamatok
- tárolt adatok ({adat}állományok)

A rendszerelemző feladata a fentebbi jellemzők meghatározása.

Egy szervezeten belül további (rész)rendszereket lehet felismerni, és hasznos ha a rendszerelemző tudatában van ezek létezésének, még akkor is ha ezek részletekbe menő leírása nem a feladata.

- **Funkcionális Rendszer** a szervezetnek az a részrendszere, amelynek keretében végzik az emberek azokat a tevékenységeiket, amely a szervezet céljainak megvalósítását szolgálja. Ez az a részrendszer, amely számítógéprendszer tartalmazhat, azonban az emberi tevékenységnek ez a számítógéprendszer csak egy kis részét fedi le. Más egyéb segítő adminisztratív, emberi rendszerfelületekről kell gondoskodni, amelyek lehetővé teszik, hogy egy számítógép rendszer helyesen működjék (vagy hogy egyáltalán működjék).
- **Szociológiai Környezet** az emberek értékeivel, értékrendszerével foglalkozik. Mit tartanak jónak, rossznak egyenrangú vagy magasabb rangú kollégáikkal való kapcsolatukban. Egy számítógép rendszer bevezetése erőteljesen befolyásolja ezeket a viszonyokat, a munkaköröket és a hatás - és feladatköröket.
- **Politikai Rendszer** az emberek jólétével, szociális problémáival foglalkozik. Milyen a szakszervezetek egymás közötti kapcsolata és viszonya a vezetéshez? A számítógép hatása a személyes érdekekre, a karrier elképzelésekre, a szervezeten belül elfoglalt társadalmi státuszra komoly hatást gyakorol és kritikus tényező lehet a rendszer elfogadása, befogadása, a későbbiekben pedig a biztonsága szempontjából.

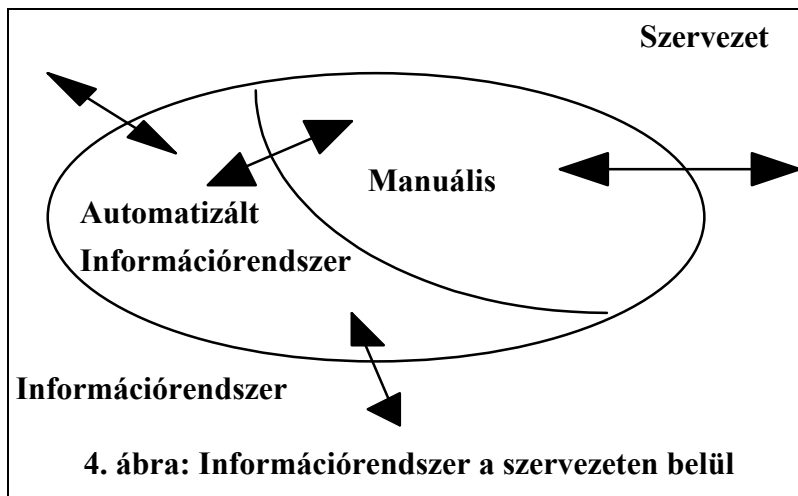
Az információk között is célszerű különbséget tenni, két legfontosabb osztálya a feldolgozandó információknak:

- **Működési információk**, azok, melyeket rutinszerűen használ a szervezet, és a napi feladatainak ellátásához feltétlenül szükség van
- **Vezetési / Irányítási információk**, azok, melyek a különböző vezetői szintek számára a döntési folyamatban fontos szerepet játszanak, és kevésbé rutinszerűek mint a működési információk.

Informatikai szempontból az automatizált adatintenzív, tranzakció-orientált rendszerek érdekesek.

- **Adatintenzív rendszerek**, azok a rendszerek, melyek az *állandó jellegű*, működési információkkal dolgoznak, amelyek a szervezet számára *közösek és integráltak*. Az állandóság itt azt jelenti, hogy egy bizonyos ideig tárolni kell még, mert ezek az adatok a szervezet folyamataival, tevékenységeivel függenek össze. Az integráció azt, hogy az adatok teljes halmaza úgy tekinthető mint több kisebb, egyébként különálló adathalmazok egyesítése, továbbá egymással összhangban vannak és nincs köztük ellentmondás. A közös adatok azt jelentik, hogy az integrált adathalmaz egyes elemeit sok különböző felhasználó együttesen használhatja.

- **Tranzakció-orientált rendszerek**, olyan rendszerek, amelyek az adatok egy adott állapotának az átalakításával (*transzformáció*) foglalkoznak, a szervezetben illetve környezetében bekövetkezett esemény hatására.



Az informatikában megkülönböztetnek **Vezetői Információs Rendszereket** (MIS, Management Information Systems), amely tulajdonképpen gyűjtőfogalma az összes alkalmazási (rész)rendszernek, valamint

Döntéstámogató Rendszereket (DSS, Decision Support Systems).

Az adatintenzív tranzakció-orientált rendszereket innentől kezdve információrendszernek fogjuk hívni a rövidség kedvéért. Néhány alapvető fogalom, amely implicit módon hozzátartozik ezekhez az információrendszerekhez a következő: az adat átalakításoknak

- *konzisztensnek* (*ellentmondásmentesnek*) kell lenniük, azaz bizonyos előírt szabályoknak meg kell felelniük az átalakítás utáni állapotoknak,
- *febonthatatlanság* azt jelenti, hogy a tranzakció vagy teljes mértékben végrehajtodik vagy egyáltalán nem,
- *tartósság*, vagyis az adat-átalakítás után az eredmény nem tehető semmissé.

Ahogy ezt már előbb megállapítottuk az információrendszerek diszkrét rendszerek, azaz diszkrét elkülöníthető állapotaik vannak, még ha nagy számú is; ezek az állapotok az adatok, adatelemek állapotainak, értékeinek kombinációjával foghatók meg. A tranzakciók ezen az adatokon okoznak állapotváltozást, a rendszert az egyik diszkrét állapotából a másik állapotba viszik át.

2. Információrendszerek fejlesztése

Machiavelli (1513):

Semmi sem lehet nehezebb, a sikere kétségesebb, sem veszélyesebb mint megtervezni és végbeinni egy új rendszer létrehozását. A kezdeményezőnek minden olyan ember ellensége, aki bármely hasznát is húz a régi rendszer megőrzéséből és csupán langyos védelemre számíthat mindazok részéről, akik az új rendszer bevezetésével nyernének.

A fentebbi idézet az új rendszerek kifejlesztésének nehézségével szembesíti a leendő rendszer készítőket. Tehát nemcsak a rendszer bonyolultsága okoz problémát, hanem más tényezők is. Ebben a szekcióban az információrendszer fejlesztéssel kapcsolatos legfontosabb fogalmakat fogjuk áttekinteni, amelyek természetesen különböző módszertanokhoz is kapcsolódnak.

2.1 Információrendszerek modelljei és ábrázolásai

2.1.1 Modellek

A leendő információrendszer részletes, megvalósítható leírását, modellekben fogalmazzuk meg. A tapasztalat azt mutatja, hogy egy lépésben - az információrendszerek komplexitása miatt -, egyetlen formalizmust használva nem lehet olyan pontossággal lefedni a rendszer összes lényeges oldalát, mely kielégítő lenne a rendszer kivitelezéséhez, tehát a bonyolultság¹ kezelésére használunk modelleket.

Az elemzés és a leendő információrendszer különböző oldalainak leírására a következő modellek használhatók:

- *szervezeti* modell, amely a munkamegosztást és a munkamódszerét, - módját írja le a valóságban; az esetleg létező Szervezeti és Működési Szabályzat, a munkaköri leírások hasznos kiinduló pontul szolgálhatnak;
- *információ* modell, ez mutatja be a rendszer információit, azok tartalmát és származtatását a valósághoz kapcsolódóan;
- *adat* modell, amely azokat a valóságban létező fogalmakat, objektumokat, entitásokat fogja össze, amelyekről eltárolnak bizonyos *információkat* és a köztük fennálló *kapcsolatokat*;
- *folyamat* modell, amelyben a való világ szerkezete, tevékenységeinek irányítási struktúrája lesz megfogalmazva.

Természetesen ezek a modellek egymással kapcsolatban állnak, explicit vagy implicit módon megkövetelik az integráltságukat, az elemeik között fennálló ellentmondásmentességet és összhangot, bizonyos kis mértékű átfedések és redundancia viszont elfogadható.

2.1.2 Ábrázolások

- *Fogalmi vagy logikai ábrázolás.* Ez az ábrázolási mód a fentebbi négy modellt absztrakt, de a felhasználók számára érthető, döntően nem műszaki szakkifejezések segítségével írja le.
 - ◆ A *szervezeti* modell ebben az ábrázolásban a szervezet tevékenységeit tartalmazza, valamint az információkhoz illetve anyagokhoz való viszonyát.
 - ◆ Az *információ* modell azokat az információkat gyűjti össze, amelyek a szervezeten belüli káosz csökkentéséhez (az entrópia növekedés fékezéséhez) vezetnek, valamint az információk forrását és származtatási / levezetési módját.

¹ Bár ennek a jegyzetnek nem tárgya a bonyolultság elmélet, de azt érdemes megjegyezni, hogy a nagy információrendszerek bonyolultsága összemérhető egy ország jogrendszerével. A szoftverek, információrendszerek azt a technológiai határt jelentik, amelyet jelenlegi tudásunk szerint még kezelni tudunk.

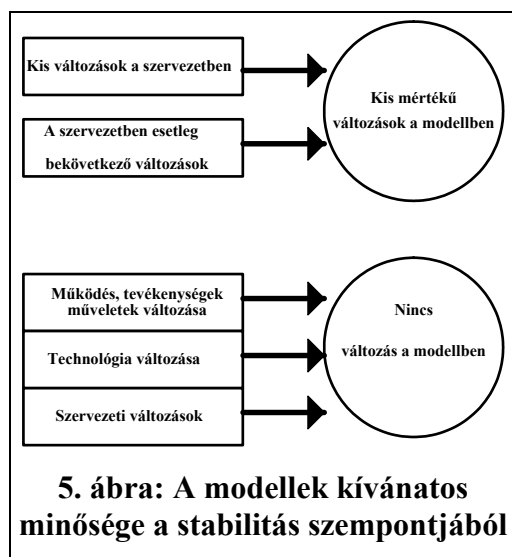
- ◆ Az *adat* modell, a szervezet tevékenységei közötti kommunikáció megvalósításához eltárolandó adatok meghatározását és az adatok között fennálló kapcsolatokat adja meg.
- ◆ A *folymat* modell a szervezeti tevékenységekben végrehajtott folyamatokat mutatja be, amelyek a szervezet működtetéséhez, irányításához szükségesek és a szervezeti valamint jogi szabályozást jelenítik meg.
- *A műszaki vagy fizikai ábrázolás.* A négy modellnek ebben az ábrázolásában a rendszernek azt a részét jelenítjük meg, amelyet számítógép fog végrehajtani vagy segíteni, és műszaki / technikai vagy fizikai kifejezéseket használnak. Azt mutatja meg, hogy a felhasználó számára, hogyan fog kinézni a rendszer műszaki vagy fizikai szempontból.
 - ◆ A *szervezeti* modell a számítógépesített tevékenységek számítógép tranzakcióit, programjait fogja ábrázolni, beleértve a szervezet telephelyeit összekötő számítógép hálózatot és az ezekhez kötődő, a felhasználók által elvégzendő, de nem számítógépesített tevékenységeket.
 - ◆ Az *információ* modell az információk megjelenítését, megjelenítési formáját mutatja be.
 - ◆ Az *adat* modell az eltárolandó adatok leírását az adatbáziskezelők műszaki nyelvén fogalmazza meg.
 - ◆ A *folymat* modell a program terveket önti formába, a bemenetek, kimenetek, és a moduláris szerkezetük értelmében. Továbbá az egymással kommunikáló számítógép folyamatokat az adott adat kommunikációt kezelő rendszernek megfelelően mutatja be.
- *A megvalósítás ábrázolása.* A négy modellnek ez az ábrázolása azt írja le, hogy a leendő rendszer hogyan valósul meg számítógépeken, terminálokon, hálózatokon, adatállományokon és adatbáziskezelő rendszereken, stb.
 - ◆ A *szervezeti* modell a tevékenységeket a szervezeti környezetbe ágyazva érzékelteti: a szervezeti egységek a jelentési kötelezettségek, az alá-fölé rendeltség hierarchiájában, a szervezeti egységek és az alkalmazottak funkcióiból származó *feladatok* (*task*) értelmében. Továbbá az összes feladatra vonatkozóan a felhasználók által követendő eljárásokat. A kommunikációs hálózat szintén ehhez az ábrázolási módhoz tartozik.
 - ◆ Az *információ* modell az információkat a programozási nyelvek formalizmusának megfelelően adja meg, azaz deklarációk formájában.
 - ◆ Az *adat* modell azt mutatja be, hogy a háttértárakon (lemez, szalag, egyéb) milyen módon tárolják el az adatok szerkezetét és a köztük fennálló kapcsolatokat.

- ◆ A *folyamat* modell az adatfeldolgozást valamilyen programozási² nyelv formájában jeleníti meg, ebbe beleértve az adatkommunikációt.

2.1.3 A modellek stabilitásának elemzése

Definíció 2-1 Stabilitás

A modell stabilitása azt jelenti, hogy a modell képes legyen a kívülvilágban bekövetkező megrázkódtatásokra, zavarokra, kiszámíthatatlan jelenségekre úgy reagálni, hogy azokat befogadja, igazodjon hozzájuk (adaptivitás). (5. ábra)



A stabil modell olyan, hogy a jövőben nincs szükség indokolatlan módosítására, a karbantartás hatásának inkább mérséklőnek, fékezőnek mintsem felerősítőnek kell lennie. A stabilitás elemzésnek a célja az, hogy megerősítse azt, hogy a szervezet életében bekövetkező kis változások legfeljebb a modell csekély változtatásához vezetnek. (Bizonyos változások a szervezetben nem követelik meg a modell megváltoztatását, de ezt nem várhatjuk el minden esetben, kis mértékű változtatásokat el kell tudni viselni.)

A szervezet modelljének nem szabad sem az alkalmazott technológiából származó korlátozásokat tartalmaznia, sem az eljárásrendből és a szervezet felépítéséből származókat, függetlennek kell lennie az osztályok közti felelősség felosztástól is. Így bármilyen változás az előbbieken nem vezet a modell megváltozásához. Ha például a modell feltételezi, hogy a befizetések postait csekken érkeznek, akkor a modell elavulttá válhat, amikor a szervezet áttér az elektronikus pénzáttalásra.

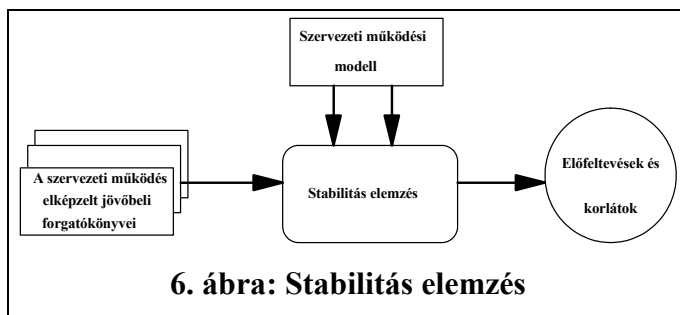
A stabilitás elemzés a következőket jelenti (6. ábra). Az elképzelt jövőnek megfelelő forgatókönyveket készítenek a szervezet működéséről: a szervezeti célkitűzések módosulása, a működési szabályok változása, a külső környezet változásai (beleértve a lehetséges törvény módosulásokat, új törvényeket). Ezeket összevetik a szervezet modelljével és megvizsgálják, hogy a modell mennyire tud megbirkózni ezekkel a hipotetikus változásokkal.

Az egyes forgatókönyvek bekövetkezésének valószínűségét is meg kell becsülni. Sokszor hasznos ha a modellt úgy módosítják, hogy az képes legyen alkalmazkodni a azokhoz a helyzetekhez, amire számítanak. Természetesen nem lehet az összes lehetséges forgatókönyvre felkészülni (azaz előre gondoskodni a lehetséges problémák leküzdéséről). A stabilitás elemzés így egyenesen elvezet a szervezet modelljének felülvizsgálatához és ugyanakkor kiemeli és láthatóvá teszi a modellben benne levő előfeltevéseket és korlátokat. Ezeket a felismert feltételeket és korlátokat közölni kell a vezetéssel azért, hogy figyelembe tudják venni a tervezésnél.

² A programozási nyelvek bármelyik generációjáról szó lehet (3GL, 4GL).

De magától értetődő módon a stabilitás elemzés akár a többi ellenőrzési, megerősítési feladat és maga a modellezési tevékenység is egy végtelen feltáró, felfedező jellegű munka és nem pedig egy véges lépéssorozatból álló konstrukciós feladat. Vagyis más szavakkal, bármennyit is dolgozik a rendszerelemző mindig van lehetőség egy kicsit többet csinálni, és lehetséges, hogy még mindig rejtve maradt valamilyen nagyon fontos tény. Ezért annak a meghatározása, hogy mikor fejezzék be az elemzést sokkal

inkább vezetői döntés semmint műszaki, informatikai.

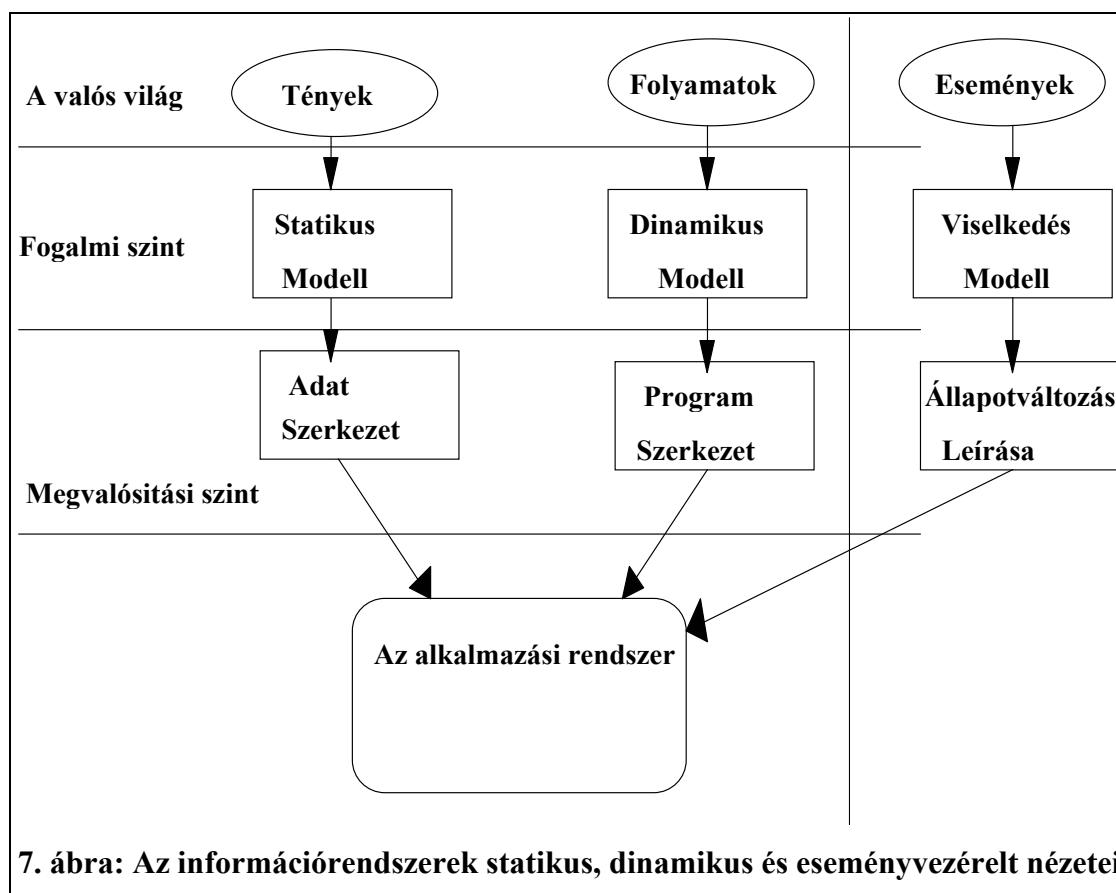


2.1.3.1 A JÖVŐ MODELLEZÉSE

A gazdasági racionalitás és a rugalmasság között ésszerű kompromisszumot kell kötni - a jelenlegi követelményekhez történő igazodás csorbíthatja az adaptációs képességet a jövőbeli követelményekhez. Itt

fellép egy időzízési probléma, a lehetséges jövőbeli követelményeknél, mind a funkcionális mind a nem-funkcionális követelményeknél. Ha valamit most nem kérnek, de lehet, hogy szükség lesz rá a jövőben, kell-e modellezni? Vagyis érdemes-e a modellünket most elbonyolítani azért, hogy bizonyos módosításokat később könnyebbé tegyünk, például előre menjünk az adatok mennyisége növekedésének.

Megalapozott döntéshez szükség van a módosítások költségére. Relációs adatbáziskezelő alkalmazásával, vagy modern CASE eszközök használatával könnyebb újabb objektumokat a modellhez hozzáilleszteni mint a meglévő modellben fennálló szemantikai és összhangot garantáló kapcsolatok jelentését megváltoztatni. Ugyancsak tudni kell, hogy mi a valószínűsége a leendő követelmények megjelenésének, milyen gyakran fordul ez elő, és mennyire pontosan lehet ezt előre specifikálni? (Ha valószínűtlen, távoli és homályos, felejtsek el!)

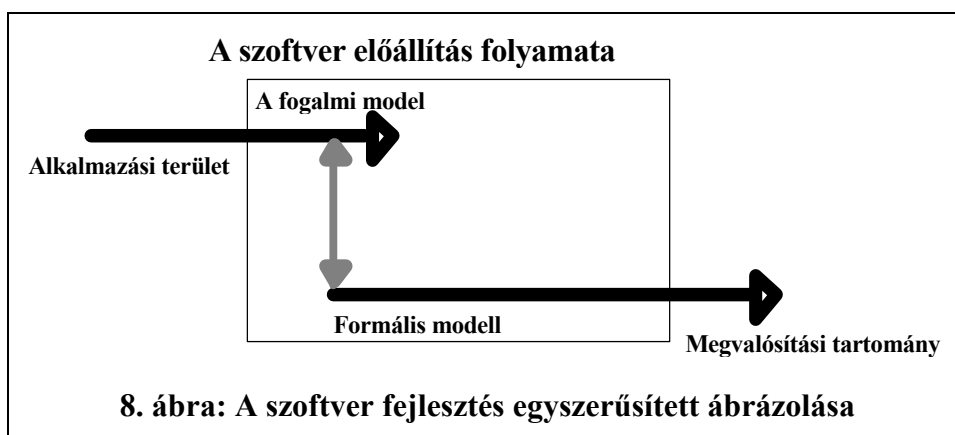


2.1.4 Az információrendszerek fejlesztésének főbb elvei

Az információrendszerek fejlesztésének egyik lehetséges módja az életciklus megközelítés (ld. 2.3.1) a másik a prototípuson alapuló evolúciós. Gyakran használják a prototípusokat az életciklus modelleken alapuló fejlesztések kiegészítéseként, ezért nem zárjuk ki teljesen a prototípusra alapuló módszereket, de ebben a jegyzetben főleg a strukturált, életciklus modellekre koncentrálnunk. A fejlesztés két nagy fázisát különböztethetjük meg:

A részletekre bontás szakasza, amikor a rendszert felbontjuk értelmes részrendszerekre, ezek kapcsolatát elemezzük és leírjuk.

A szintézis szakasza (kivitelezés), amikor a rendszert alkotó részeket összekapcsoljuk, szintézis útján egy új rendszert hozunk létre.



Az információrendszereket felfoghatjuk tehát úgy, mint a valóság egy szeletének az ábrázolását, az adott szervezet egy darabjának a leírását, vagyis a létező tények és a valóságban végbemenő folyamatok megfogalmazásának. Ezért az információrendszerek fejlesztése tulajdonképpen modellezési, modell leírási probléma.

Ezeket a modelleket besorolhatjuk két vagy három főbb kategóriába.

A rendszerek dinamikus oldalának lehet tekinteni folyamatokat, tevékenységeket, adatfolyamokat, tranzakciókat. Ezt a nézetet a használat nézőpontjának nevezhetjük, azaz azt tükrözi, hogy *'hogyan segítik a szervezet működését'*. A statikus oldalhoz sorolhatjuk a tárgyakat, objektumokat, kategóriákat és a köztük fennálló kapcsolatokat, azaz azokat, *'amik segítik a szervezet működését'*. Egyre terjed az a felfogás, hogy érdemes külön venni és megkülönböztetni a szervezeti eseményeket, az ezek által kiváltott informatikai eseményeket, amelyek valamilyen változtatást, átalakítást kezdeményeznek az információrendszerben, és az események által beindított folyamatokat. Ezt az oldalt a rendszer viselkedésének nevezhetjük, vagyis az eseményekre adott rendszer választ tükrözi, vagyis *'hogyan reagál a rendszer a szervezet környezetében bekövetkező eseményekre'*.

A fentebbi ábrán (8. ábra) az automatizált, számítógépes információrendszerek szoftver fejlesztés folyamatára egy sematikus ábrázolását mutatjuk be. Ez az ábrázolás rámutat annak a ténynek a fontosságára, hogy először egy olyan modellt kell kialakítani, amely az alkalmazási terület megértését tükrözi vissza (vagyis a jellegében kognitív) azután egy sor átalakításon keresztül egy sokkal formálisabb modell sorozatot alakítanak ki, amelyek megfogalmazását már befolyásolják a fizikai megfontolások.

2.2 Módszertanok a gyakorlatban

Az információrendszerek (iparszerű) készítésének nagyon fontos segédeszközei a szoftveréletrajz több fázisát, illetve egyes fázisait támogató módszertanok. Hosszú évek kutatásai/fejlesztései során alakultak ki a jelenlegi módszertanok jellemzői az ún. strukturált módszerek alkalmazása, és a különböző diagramtechnikák, amik közérthető ábrák felhasználásával jelentősen megkönnyítik a műszaki/technikai szakemberek és végfelhasználók közötti kommunikációs szakadék leküzdését.

Információrendszerek fejlesztése

Egy felmérés szerint az Egyesült Államokban, Németországban, Nagy-Britanniában és Franciaországban a rendszerelemzési technikák nagymértékben elterjedtek. A felmérésben szereplő fejlesztésekben az ilyen technikák országokénti elterjedtségére a következő adatokat kapták:

| | |
|----------------|-----|
| Nagy-Britannia | 33% |
| Franciaország | 28% |
| Németország | 16% |
| USA | 17% |

Az önkormányzatokra és az országos kormányzati szférákra vonatkozó megfelelő adatok a következők:

| | |
|----------------|-----|
| Nagy-Britannia | 50% |
| Franciaország | 20% |
| Németország | 0% |
| USA | 50% |

Nagy-Britanniában, Franciaországban és Hollandiában egyértelműen vezető szerepet játszanak a kormányzati szabványok, nevezetesen az **SSADM**, a **MERISE** és az **SDM**. Vagyis nemcsak az országos és helyi kormányzatokban folyó rendszerfejlesztésekben, hanem más alkalmazási területeken is használják ezeket a módszereket, például pénzügyi és biztosítási rendszerek, ipari gyártórendszerek, kis- és nagykereskedelem, stb. Egy felmérés szerint az **SSADM** (40%+), a **MERISE** (45%), Hollandiában az **SDM** (40%) részesedést el.

A strukturált rendszerelemzési / - fejlesztési technikákat döntően nagy és közepes projektekben használták, de néhány kisebb munkában is. A fejlesztői csoportok létszáma 30 főnél kevesebb volt a felmérésben szereplő rendszerek 25%-nál.

Strukturált rendszerelemzési / - fejlesztési technikákat alkalmazók több mint fele használt valamilyen CASE eszközt.

Ezek a módszertanok nagyon sokat segítenek az ún. szoftver krízis leküzdésére, amit a következő statisztika illusztrál:

A fejlesztők, cégek által valamilyen formában átadott alkalmazás fejlesztések, szoftverrendszerek használatba kerülésének statisztikája a Pentagonnál:

| | |
|-----|---|
| 47% | leszállított, soha nem használt |
| 29% | kifizetett, de soha le nem szállított |
| 19% | átdolgozott, vagy kidobott |
| 3% | használták a változtatások elvégzése után |
| 2% | volt úgy használva, ahogy leszállították |

(forrás: U.S.A: hadseregének statisztikája)

A hardver elveszti meghatározó szerepét. A legutóbbi időkig a kiválasztott hardver jellemzői határozták meg a rendszerfejlesztés irányelveit. Azaz a hardverrel kapcsolatos költségek diktálták a feltételeket, aminek eredménye gyakran az volt, hogy a felhasználói követelményekből kellett engedni. Az elmúlt évek hihetetlen

gyorsaságú hardver áresése következtében a hardverrel kapcsolatos szempontok nem játszanak elsődlegesen meghatározó szerepet, nem akadályozzák és nem is korlátozzák a megfelelő megoldás kialakítását.

- A hardver várható élettartama jelenleg 5 év,
- egy alapszoftveré (operációs rendszer, stb.) 10 év,
- alkalmazást készítését segítő szoftver (adatbáziskezelő, stb.) 15 év,
- egy információrendszeré 30 év .

(forrás: Guidelines for an informatics architecture - Commission of the European Communities).

Új követelmények az alkalmazásokkal szemben. Az információrendszerek fejlesztőinek újabb és újabb alkalmazási területekkel kell szembenéznük, mint például az irodaautomatizálás, döntéstámogató rendszerek és szakértő rendszerek. Ezenkívül a döntési folyamatok jó előkészítése, megalapozása felkeltette az igényt az integráltság iránt, ami pedig az információrendszerek adatainak megosztására, konkurens elérésére és ellentmondás-mentességük fenntartására világított rá.

Felhasználók elégedetlensége. Az információrendszerek fejlesztésével kapcsolatos problémák leginkább a felhasználókkal való együttműködés során jelentkeznek, a végső eredménnyel azonban a felhasználók sokkal többször elégedetlenek mint elégedettek. Az olyan jelenségek mint pl. a késedelmes leszállítás, költségtúllépés, rugalmatlanság és a nem megbízható rendszerek, amelyek rendszeresen visszatérő felhasználói panaszoknak számítanak.

A hagyományos információrendszer-készítés hátrányait a következőkben összegezhethetünk:

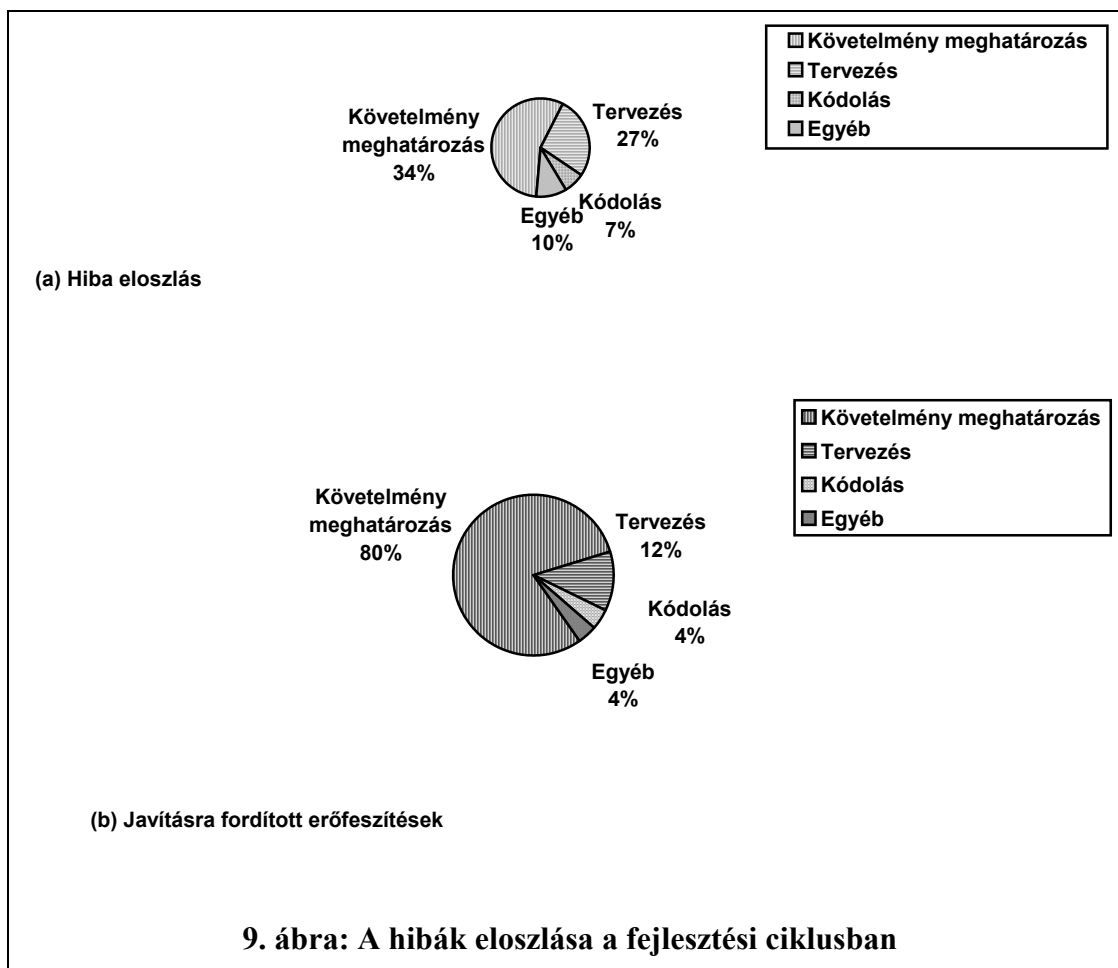
- gyenge követelményspecifikáció, ami jelentős mértékben a szöveges leírásokban elkerülhetetlenül előforduló kétértelműségek következménye,
- gyenge rendszertervek, ami annak a következménye, hogy nem voltak szabályok, heurisztikák, ökölszabályok a követelmény specifikáció átalakítására jó rendszertervvé,
- a rendszer nehezen volt karbantartható,
- exponenciálisan egyre növekvően többet és többet kellett a rendszerek karbantartására fordítani a gyenge tervezés következtében, emiatt az új alkalmazásokra egyre kevesebb erőforrás jutott, ami az új felhasználói követelmények növekvő hátralékában jutott kifejezésre, azaz nagyon sok felhasználói kívánságnak, kérdésnek semmilyen esélye sem volt arra, hogy valaha is megvalósul.
- a fejlesztést nehéz volt formális minőségbiztosítási eljárásrendbe illeszteni.

Egyes felmérések azt mutatták a 80-as évek elején, hogy a számítástechnikában alkalmazottak 92%-a foglalkozik a rosszul tervezett rendszerek tesztelésével és hibakereséssel.

A fentebbi okok együttesen vezettek oda, hogy az információrendszerek 'informális', hagyományos fejlesztése alkalmatlan a jelenlegi információrendszerekkel szemben

fennálló igények kielégítésére és ezért szükség van tudományosan megalapozott rendszerfejlesztési módszertanok kialakítására.

Bár jelentős javulás állott be a legutóbbi időkig az informatikai és szoftver ipar fejlődése következtében, különösen az alkalmazott vezetési és irányítási módszerek fejlődése miatt. Azonban a fejlesztési projektek sikeressége még mindig gyengébb



más iparágak teljesítményéhez viszonyítva.

Több jelentős piackutató, az informatikai fejlesztésekben érdekelt tanácsadó cég végzett felmérést a közelmúltban. Ezek az eredmények továbbra is elgondolkodtatóak:

| Helyszín | Év | Forrás | Minta darabszáma | Sikeresség % |
|-------------------|-----------|----------------|------------------|--------------|
| London, Britannia | Nagy-1990 | Kearney | 400 | 11 |
| London, Britannia | Nagy-1994 | Pagoda | 100 | 11 |
| Dennis, USA | 1995 | Standish-kicsi | 365 | 28 |
| | 1995 | Standish- | | 16 |

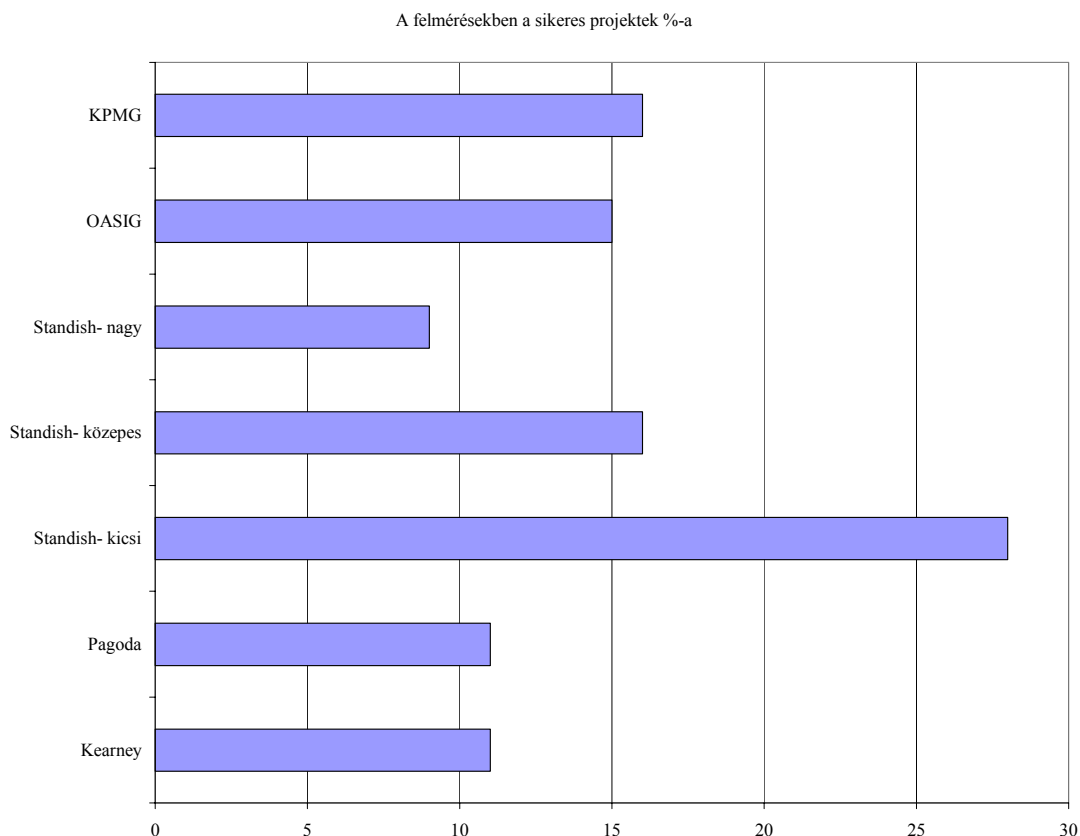
| | | | | |
|---------------------------|------|---------------|-------------|-----------|
| | | közepes | | |
| | 1995 | Standish-nagy | | 9 |
| Sheffield, Nagy-Britannia | 1996 | OASIG | 45 | 15 |
| Toronto, Canada | 1997 | KPMG | 176 | 16 |
| Mindösszesen | | | 1086 | |
| Sikeresség átlaga | | | | 15 |

1. táblázat A felmérésekből készített összegzés

A fentebbi adatok segítenek felbecsülni, hogy a kormányok, a tudomány, a kereskedelem és az ipar területén végrehajtott informatikai (szoftver) fejlesztések milyen arányban bizonyultak sikeresnek a 90-es évek folyamán.

Ennek a táblázatnak a helyes értékeléséhez azonban figyelembe kell venni a következőket:

- Hét forrás adott „sikerességi %” mértéket;
- A felmérési adatok három országra terjednek ki;
- Nagyjából a 90-es évekből 7 évet fog át a felmérés;
- A felmérést öt különböző szervezet végezte el és hajtotta végre.

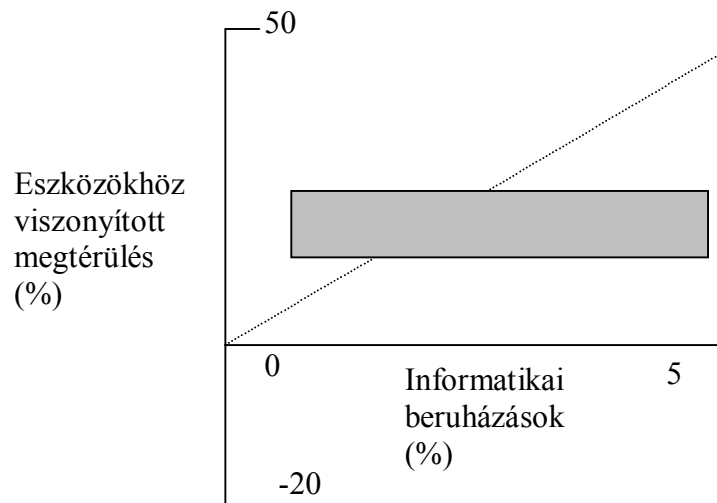


10. ábra: A felmérésekből származtatott projekt megvalósulások sikerességi százaléka

2.2.1.1 AZ INFORMATIKAI BERUHÁZÁSOK HASZNA – AVAGY A STRASSMAN TÉNYEZŐ

Strassman³ kutatásai arra irányultak, hogy vizsgálja az informatikai befektetések megtérülését a vállalat vagyonához (vállalati eszközökhöz) viszonyítva. A pontozott vonal, azt érzékelteti, amit általában józan paraszti ésszel várnánk el az informatikai beruházások megtérülésétől: a magasabb informatikai beruházások egy megfelelően magasabb versenyelőnyt biztosítanak a piacon, és ezzel magasabb megtérülést nyújtanak az adott vállalkozásnak. Azonban nem ez, amit Strassman talált. Néhány száz vállalatot vizsgált meg az informatikai beruházásokon mérhető megtérülés szempontjából. Az eredményeket a felvitte egy diagramra (11. ábra) pontokkal jelölve az egyes értékeket. Azt tapasztalta, hogy a pontok döntő többsége a szürke téglalap területére esett, ami azt jelezte, hogy a vállalati eszközökhöz viszonyított megtérülés viszonylag állandó értéket mutat és független a befektetések nagyságától.

³ [Strassmann97], [Strassmann97a], [McAteer95]



11. ábra Az informatikai beruházások és az eszközökhöz viszonyított megtérülés (Strassman nyomán)

A másik érdekes oldala ennek, hogy a szürke téglalap alakja két teljesen különböző mintavétel esetén is lényegében ugyanaz maradt, az egyik adatgyűjtést a 80-as években, a másikat a 90-es években hajtotta végre. Vannak akik vitatják ezeket az eredményeket, de az informatikai fejlesztésekről szóló döntéseknél azért érdemes ezt figyelembe venni. Informatikai beruházások, információrendszer fejlesztések eredményessége a következő tényezőkkel mutatnak statisztikai összefüggéseket:

- A cég piaci értéke megmarad nem csökken, a részvények értéke kissé nő, vagy nem csökken.
- Megőrzi a cég a piaci részesedését. Az új technológiát korán alkalmazók esetleg növelni tudják a piaci részesedésüket.

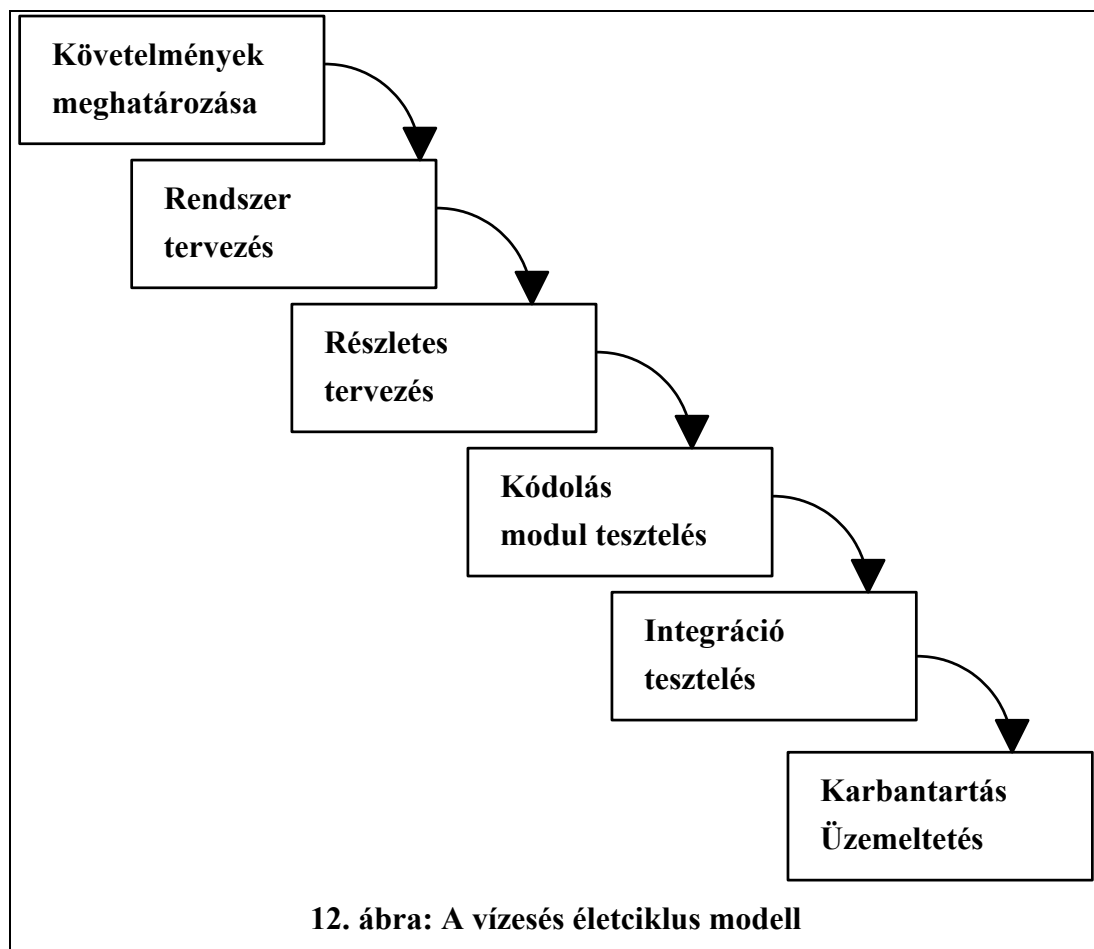
2.3 A fejlesztés szakaszai

2.3.1 Életciklus modellek

Nagyon röviden ismertetjük az elfogadott rendszeréletciklus modelleket.

A "Vízesés" életciklus modell:

- Az alkalmazás követelményeinek meghatározása
- Rendszer tervezés
- A részletes tervezés
- Kódolás és hibakeresés, modul tesztelés
- Integrációs tesztelés
- Működtetés és karbantartás



A modell alkalmazása:

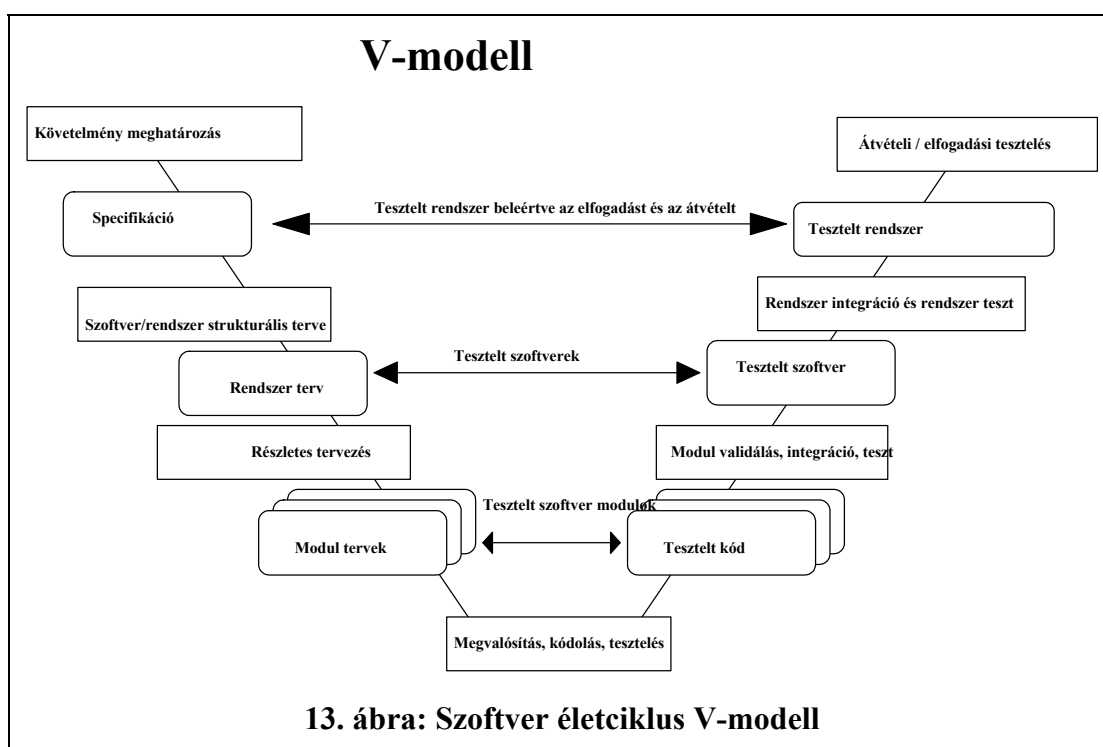
- Hagyományos - bármely szakaszból bármely másikba vissza lehet lépni
- Szoftver mérnöki - visszalépés csak a ciklus legelejére lehetséges
- Prototípus - A követelmény specifikáció és a részletes tervezés között ciklikus visszalépés
- CASE - A fejlesztés minden lépésébe bekerül

A V-modellnek van néhány előnye a vízesés modellel szemben (lsd. 13. ábra.):

- az egyik szakaszban létrehozott termékek közvetlenül átkerülnek a rákövetkező szakaszba;
- azokat a termékeket, amiket le kell ellenőrizni, tesztelni a különböző szakaszokban, eltérő szintű hibaellenőrzési eljárásokkal kapcsolja össze, világosan és expliciten kijelölve ezeket a szinteket.

Az ábrán a lekerekített sarkú téglalapok utalnak az elkészítendő termékekre, a téglalapok a szakaszokra. Az idő előrehaladását úgy lehet képzelni, hogy fentről lefelé haladva a legszélső baloldali elemtől a legszélső jobboldali elemig haladunk.

A bonyolult rendszerek, szoftverek kifejlesztésére alternatív életciklusként dolgozta ki Boehm a spirál életciklust (ld. 14. ábra). Ezt a spirált legcélszerűbb egy polár koordináta rendszerben felfogni, ekkor a radiális irány mutatja a fejlesztésben addig felmerült, akkumulált költségeket, a sugár által bezárt szög illetve az ezáltal meghatározott szektor pedig a fejlesztés előrehaladását reprezentálja. A modell szerint



a spirál minden egyes ciklusa a fejlesztés előrehaladását jelenti, lényegében ugyanazokat a fejlesztési lépéseket ismételve, de természetesen minden egyes körben az előzőleg létrehozott termékek finomítását, finomabb kidolgozását hozva létre a termékek minden részére és kidolgozottsági szintjére vonatkozóan. A leendő rendszer működését globálisan leíró dokumentumból kiindulva egészen az egyes programok lekódolásáig.

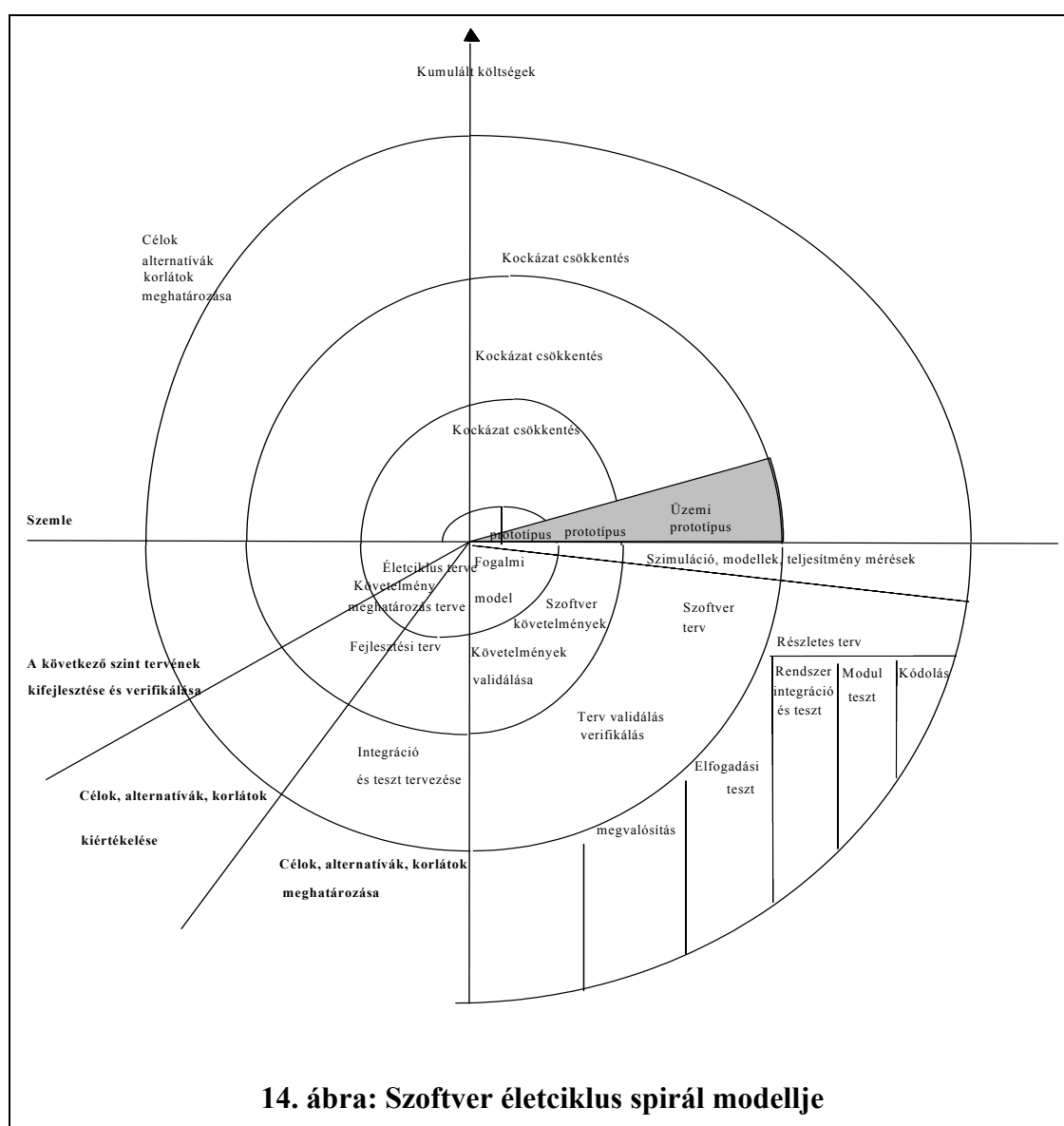
Minden ciklus a spirál balszélső legfelső negyedében kezdődik a következők meghatározásával:

- a végtermék(ek) mely részét és milyen célból kell kidolgozni (teljesítmény, funkcionalitás, adaptivitás azaz a változásokhoz történő rugalmas alkalmazkodás képessége, stb.);
- a végtermék(ek) ezen részének megvalósíthatóságának elemzése, alternatív módok elemzése (alternatív tervek, újra-felhasználhatóság, beszerzés, stb.);

- az alternatívák korlátai és hatása az alkalmazásra (költségek, ütemtervek, kapcsolatok és felületek más alkalmazásokhoz, stb.).

A következő lépés az alternatívák kiértékelése tekintettel a célokra és a korlátokra. A bizonytalanságra, instabilitásra utaló jeleket fel kell ismerni, mivel ezek a projekt kockázatát nagy mértékben fokozzák, ezért olyan költségtakarékos megoldásokat kell találni, amelyek a felbukkanó kockázatok kezelésére alkalmasak, mint például a prototípus készítés, szimuláció, felhasználók intenzív bevonása, elemző / analitikus modellezés, vagy különböző kombinációi ezeknek és esetleg egyéb más kockázat csökkentős technikáknak.

Miután a kockázatokat kiértékeltek, a felismert kockázatok viszonylagos egyensúlya fogja a következő lépést meghatározni, pl.:



- evolúciós fejlesztés (prototípus), a végtermék globális tulajdonságait minimális befektetéssel specifikálják, megtervezik a prototípus fejlesztés következő lépését,

majd egy sokkal részletesebb prototípust fejlesztenek ki, mellyel a fontosabb kockázatokat mérséklük;

- inkrementális fejlesztés, azaz a klasszikus életciklus fejlesztés a következő szakaszra, vagyis az előző szakasz eredményeire támaszkodva annak bővítését hozzák létre (fogalmi modell, követelmény meghatározás, strukturált tervezés, stb.);

A specifikáció készítés minden lépését egy *validációs* és *verifikációs*⁴ lépés követi, majd a következő (spirál)ciklus terveinek az előkészítése. Minden dokumentum kifejlesztésének utolsó szakaszát a jobboldali legalsó negyed mutatja.

Minden ciklus végén egy átfogó *szemlé*t hajtanak végre, ez jellemző tulajdonsága a spirál modellnek, az ábrán pedig a baloldali tengely érzékelteti ezt a tevékenységet. Ez a szemle az előző ciklusban előállított összes terméket vizsgálja, beleértve a következő ciklus ütem - és erőforrás-tervét. A szemlézésben az összes érintett és érdekelt fél, felhasználó, szervezet részt vesz. A másik fontos tulajdonsága ennek a modellnek, hogy nagy hangsúlyt helyez az esetleg megjelenő kockázatok összes aspektusának kezelésére és alkalmas döntési mechanizmus kialakítására.

Ezt a modellt különösen szeretik használni:

- Gyors fejlesztéseknél (Rapid Application Development, DSDM⁵, Dynamic System Development Method);
- Prototípus alapú, evolúciós megközelítésekénél;
- Szakértő és ismeretbázisú rendszereknél;
- specifikáció orientált, automatikus transzformációt alkalmazó, szimulációt használó megközelítésekénél
- Inkrementális fejlesztéseknél.

2.3.2 Információrendszer adaptációk készítésének szakaszai

Az előző fejezetek néhány szoftver fejlesztési életciklus modellt ismertettek röviden. Ezekből is látható volt, hogy számtalan szakaszolási módja van a fejlesztésnek. Az információrendszerek fejlesztése a konkrét szoftver fejlesztésnél még nagyobb területet fog át, így itt is módszertanonként különböző szakaszolással találkozhatunk:

- információrendszerek stratégiai tervezése,
- rendszerelemzés,

⁴ Validálás azt jelenti, hogy vajon a specifikáció illeszkedik-e a követelményekhez, a közölt igényekhez. A verifikáció azt jelenti, hogy az adott terméket önmagában vizsgáljuk, hogy jó-e. Pl. egy másodfokú egyenlet megoldására alkalmas programot specifikálunk és ellenőrizzük, hogy jól működik-e, ha igen akkor a verifikáció sikeres volt. Azonban lehet, hogy egy az inflációt figyelembe vevő költség eszkaláció számításra van szükségünk, ebben az esetben a validálás sikertelen, mert a követelményeinket nem elégíti ki a specifikáció.

⁵ <http://www.dsdm.org/basics.htm>

- rendszertervezés,
- rendszerkészítés.

Az Euromethod (ld. [CCTA95B], [Turner96], [Euromethod94]) ezt összefoglaló névvel információrendszer adaptációnak nevezi (IR-adaptáció információrendszer: adaptáció).

2.3.2.1 INFORMÁCIÓRENDSZEREK STRATÉGIAI TERVEZÉSE

A szervezet tevékenységét működését elemzi, de nem olyan részletességgel, mint amikor egy konkrét rendszert akarunk megtervezni, amely egy adott tevékenység egészét vagy annak egy részét segítené. Ekkor a már létező rendszerek elemzésére is szükség van, abban az értelemben, hogy milyen hasznot hajtanak, nyújtanak-e valamilyen előnyt a szervezetnek. Az információrendszerek stratégiai tervének meg kell jelölnie azokat a rendszereket, amelyeket létre kellene hozni és azt a sorrendet, amelyben a kivitelezésük megtörténhetne. A rákövetkező szakaszok a szervezet illetve a működés, a tevékenységek egyre szűkebb körére koncentrálnak. Ennek a szakasznak a végterméke, a megrendelőnek *átadandó terméke*, a leendő információrendszerek terveinek egy portfóliója, ezt tulajdonképpen *tervezési terméknek* tekinthetjük.

2.3.2.2 RENDSZERELEMZÉS

Ez a szakasz a szervezet egy meghatározott működési területének helyzetét vizsgálja meg és egy helyzetfelmérési tanulmányt készítenek. A létező információrendszereket tanulmányozzák, akár manuális akár automatizált rendszerről is legyen szó. Elemzik, hogy valójában mit is csinálnak a szervezetben, és tulajdonképpen mit kellene csinálni ahhoz, hogy egy sokkal fejlettebb információrendszer működését támogassák.

Ebben a szakaszban megint keletkezik egy a megrendelőnek *átadandó termék*, amit ugyanakkor a *rendszerelemzés termékének* tekinthetünk. Ez a szakasz tulajdonképpen leíró és nem előíró jellegű.

2.3.2.3 RENDSZERTERVEZÉS

A rendszertervezési szakasz egy a leendő információrendszer vonatkozó előírást állít elő, általában elektronikus formában. Az alkalmazási terület kiterjedése sokkal szűkítettebb, mint a megelőző rendszerelemzési szakaszban vizsgált területé.

Gyakran a tervezési szakasz eredményeként megjelenő *tervezési termék* független a rendszer létrehozása során használandó eszközöktől. Azonban sokszor már ekkor lehet tudni, hogy mik lesznek a készítés során használt eszközök és ezeknek a tulajdonságait figyelembe lehet venni, különösen teljesítmény tervezési szempontból.

2.3.2.4 RENDSZERKÉSZÍTÉS (LÉTREHOZÁS)

A rendszerkészítési tevékenység tulajdonképpen nagymértékben függ a rendelkezésre álló hardver és szoftver környezettől. A rendszerfejlesztési környezet általában a következő eszközöket tartalmazhatja:

- adatbázis-kezelő rendszer;
- adatszótárak, repozitóriumok;
- képernyő tervező eszközök, grafikus felhasználói felület tervező eszközök;

- tranzakció feldolgozó eszközök;
- programozási nyelvek;
- alkalmazás generátorok.

A szoftver fejlesztési környezet kiválasztása ideális esetben a rendszertervezési szakasz befejezése után történik meg, azaz miután a rendszert minden részletére kiterjedően már megtervezték.

3. Folyamatok elemzése

A következő fejezetben a folyamatok elemzésével foglalkozunk. Az idevágó fogalmak és technikák ismertetésekor az egyik jelöléstechnikát választjuk ki, de ez a jelöléstechnika csak a külső megjelenése ugyanazoknak az elveknek és szemantikai tartalomnak, amit esetleg egy másik módszer kissé különböző jelöléssel jelenít meg.

3.1 Bevezetés a folyamatmodellezésbe

A folyamat modellezés főbb céljai:

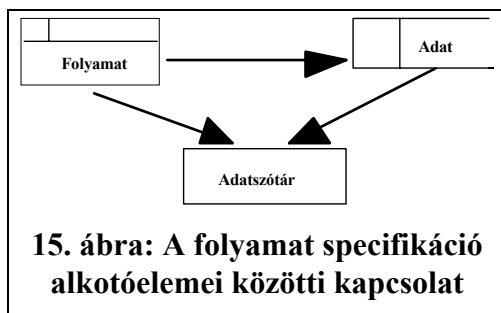
- kielégítő pontossággal írja le a jelenleg működő rendszert;
- ábrázolja azt az absztrakt rendszert, amely már nem tartalmaz fizikai korlátokat;
- visszatükrözi a leendő, igényelt rendszerrel szemben támasztott követelményeket, az elképzelt rendszer működését.

Ezeket a rendszer ábrázolásokat az interjúkból, kérdőívekből, a mérési adatokból és a dokumentumok feldolgozásából absztrakció útján hozzák létre.

A legfontosabb analízis technikák, amelyeket érinteni fogunk:

- adatfolyam diagram (Data Flow Diagram, DFD), amely a folyamatok és a közöttük levő kapcsolatokat az adatfolyamok értelmében ábrázolja;
- adatszótár, amely az adatfolyamokat alkotó adatelemek leírását tartalmazza;
- folyamatok specifikációja, amely a folyamatok írja le részletesen.

A fentebb felsorolt modellezési elemek közötti kapcsolatot a következő ábra érzékelteti (15. ábra).



Az információrendszerek középpontjában⁶ az *adatok* vannak, ennek megfelelően a folyamata specifikációban centrumában is áll. Ezeknek az adatoknak általában jól definiált szerkezete van; az adatok leírását az *adatszótár* tartalmazza, nevezetesen: az adatelemek tulajdonságait, jellemzőit valamint megjegyzéseket, az analízis során feltárt és tapasztalt dolgokról. A rendszeren belül tárolt

⁶ lásd 1.3 fejezet 9. oldal.

adatokat a *folyamatok* módosítgatják, változtatják az információrendszer működési céljának megfelelően.

3.1.1 Bevezetés az adatfolyam modellezésbe

Definíció 3-1 Adatfolyam modell

Az adatfolyam modellezési technika az információrendszerek folyamatainak kapcsolatát írja le a köztük áramló adatfolyamok értelmében, feltüntetve a rendszerben tárolt adatokat is.

A fentebb megfogalmazott modellezési célkitűzéseknek sok jelölés megfelel, több alternatív diagram technika terjedt el, a CASE eszközök is különbözőket támogatnak. az alábbiakban bemutatunk néhányat közülük, majd arra fogunk koncentrálni, amely a további tanulmányokat előkészíti és segíteni fogja (16. ábra).

Az ábrán látható jelölések közül az SSADM módszertannak megfelelőt fogjuk használni, aminek több oka van:

1. Magyarországon elfogadott kormányzati ajánlás az államigazgatásban.
2. Több Magyarországon forgalmazott CASE eszköz támogatja a módszertant és az általa alkalmazott jelöléstechnikát.
3. Nagy-Britanniában szabvány és az államigazgatásban előírt módszertan, valamint más országokban is használt módszertan.
4. Az informatikával illetve információgazdálkodással foglalkozó hallgatók részletesebben fogják tanulni az SSADM módszertant.

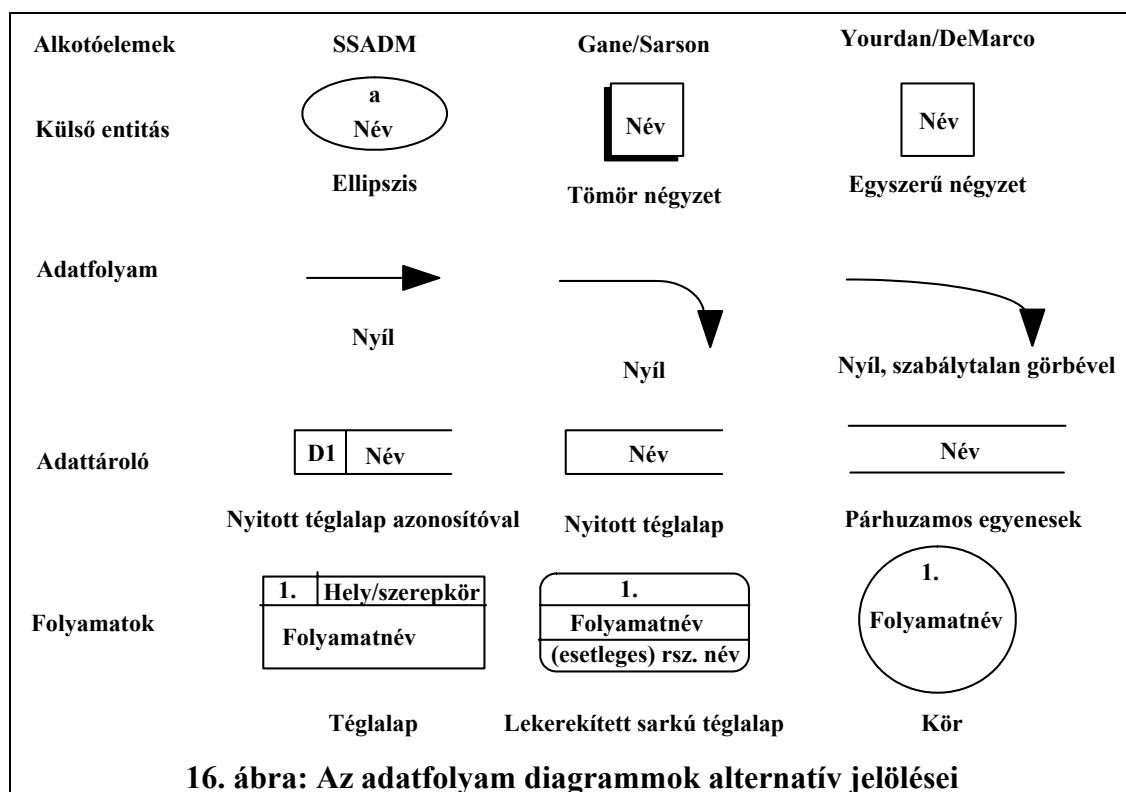
Ezért ezt a jelöléstechnikát használva ismertetjük az adatfolyam modellezést, de továbbra is utalni fogunk esetleges alternatív módszerekre és jelölésekre. Azonban a háttérben meghúzódó elméletek és elvek ugyanazok, így ez az alkalmazhatóságot és az általánosságot egyáltalán nem zavarja.

Ahogy az az ábráról is kiderül, az adatfolyam diagram legfontosabb alkotórészei a következők:

- külső entitás,
- adatfolyam,
- adattároló,
- folyamat.

3.1.1.1 KÜLSŐ ENTITÁS

Külső entitásnak vagy (információ)forrásnak és nyelőnek nevezik azokat a személyeket, osztályokat, szervezeti egységeket, szervezeteket, más rendszereket, amelyek akár információt szolgáltatnak az elemzésnek alávetett rendszernek vagy fogadnak onnan jövő adatokat. Egy külső entitás egyszerre lehet információforrás és -nyelő. Például az ügyfél, aki megrendelést ad fel és megkapja a kiszállított árut egyszerre információforrás és -nyelő. A külső entitás definíció szerint az elemzett rendszer határán kívül van, voltaképpen implicit módon a külső entitások felsorolása meghatározza a rendszer határát. Ha a rendszer határát egy falnak képzeljük, akkor az adatok a falon lévő lyukon keresztül léphetnek be a rendszerbe vagy hagyhatják el a rendszert, így a külső entitást a "lyuk" mögött levő valaminek képzelhetjük. Ez a kép azt is sugallja, hogy a rendszerhez "legközelebb" levő valamit kell külső entitásnak tekinteni, azt ami közvetlenül kapcsolatban van a rendszerrel és adatfolyamot küld be vagy fogad a rendszertől. Például egy bankfiókban az ügyfél valamilyen tranzakciót kezdeményez a folyószámláján a bank alkalmazottjánál, aki egy terminálon vagy



számítógépen valamit végrehajt. Ebben az esetben a bank alkalmazottat kell külső entitásnak tekintenünk, aki esetleg egy másik külső entitástól az ügyféltől kap valamilyen adatot.

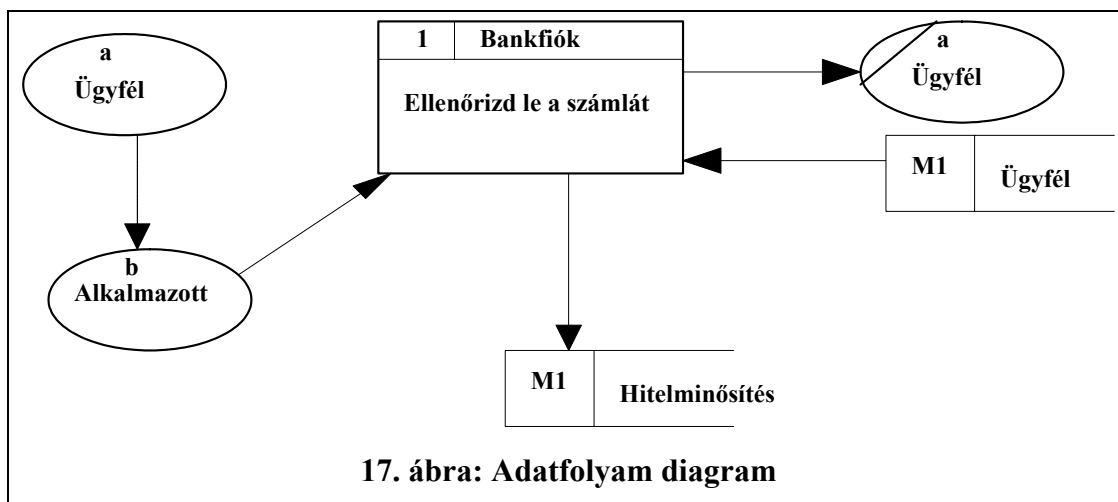
A pókháló szerű ábrák elkerülése végett, a külső entitás több példányban is megjelenhet a diagrammon, ezt egy "sapkával" jelöljük; továbbá az entitást az ABC kis betűivel azonosítjuk, valamint egy nevet adunk neki. Gyakran ez a név az elemzett terület leendő felhasználóinak feladataiból, munkaköréből származhat. De szembe kell nézni azzal az ténnyel, hogy több különböző személy, eltérő helyszíneken esetleg ugyanazokat az adatokat adja át a rendszernek, ugyanazokat a felhasználói funkciókat hajtja végre, ez a rendszerelemzőt szükségtelen absztrakciókhoz vezetheti, olyan

elnevezéseket használhat, amiket a felhasználók nem értenek meg. Ez arra mutat rá, hogy ez csak segít a felhasználói szerepkörök feltérképezésében, de nem helyettesíti a megfelelő technikát.

3.1.1.2 ADATFOLYAMOK

Ezt a nyilat - eltérően a folyamatábrákon használt nyilaktól, amelyek a vezérlésátadást reprezentálták - úgy foghatjuk fel mint egy utat, amelyen egy vagy több adatszerkezet⁷ közlekedhet föl s alá, az időpont meghatározása nélkül. Az idővel illetve ütemezéssel kapcsolatos kérdéseket a folyamatok specifikációjában részletezik. Az adatfolyam diagrammot ebben az értelemben egy vasúti térképhez hasonlíthatjuk, amelyen a vonatok által követhető utakat láthatjuk, de a menetrendet nem mellékeltek. Általában az adatfolyam egy olyan nevet kap, ami egyértelműen azonosítja azt az adatszerkezetet, amely az adatfolyamon keresztül áramlik - és értelmes a felhasználók számára.

Az adatfolyamok lehetnek egy vagy két irányúak. Különösen a két irányú adatfolyamoknál kell figyelni arra, hogy ne tartsanak vezérlési információkat vagy olvasási kérést. A programozási logika szerint sokan kísértést éreznek arra, hogy amikor olvasnak egy adattárolóból, akkor azt az ábrán egy az adattárolóba vezető nyíllal érzékeltesék, az adatfolyam adatszerkezete pedig tartalmazza a rekord kulcsát. Ez hibás! Csak az adattárolóból kivezető nyíl kell bemutatni a diagrammon ebben az esetben, mivel ez érdekes a vizsgálat és a szervezet működése



szempontjából, ez jelenti az olvasás vagy adat visszakérést.

Alkalmanként a külső entitások között is ábrázolhatjuk az adatfolyamokat, ekkor szaggatott vonalat használunk. Annak ellenére, hogy ezek a szűkebb értelemben vett

| | Külső entitás | Folyamat | Adattároló |
|---------------|------------------|----------|------------|
| Külső entitás | Külső adatfolyam | Igen | Nem |
| Folyamat | Igen | Igen | Igen |
| Adattároló | Nem | Igen | Nem |

18. ábra: Az adatfolyam ábra elemei között megengedett (adatfolyam) kapcsolatok

⁷ Az adatszerkezet az adatelemek listájával vagy struktúra diagrammal (később tárgyaljuk) adható meg.

rendszeren kívül vannak sokszor segíthetik a megértést. A táblázatban az adatfolyam diagram elemei között a megengedett adatfolyamokat tüntettük fel.

- A „Jelenlegi Fizikai Adatfolyam Diagrammon” az adatfolyamok a valóságban áramló információkat ábrázolják, vagyis a tényleges űrlapokat, formanyomtatványokat, dokumentumokat, iratokat, telefon hívásokat, stb.
- A „Logikai és az Igényelt Rendszer Adatfolyam Diagramján” ezek az adatfolyamok azokat az adatelemeket, attribútumokat jelentik, amelyeket a folyamatok használnak bemenetként és kimenetként.
- Egy folyamatban se nem keletkezhet, se nem tűnhet el adat (SSADM); dokumentumok keletkezhetnek vagy elnyelődhetnek, de kell lennie minden folyamatnál az összes bemenő adatra olyan kimenetnek, amely közvetlenül kapcsolódik hozzájuk. Ez igaz a teljes diagramra, egy folyamatra, a folyamat esetleges lebontására. Ennek az oka az, hogy a folyamatok csak az adatelemeken okozhatnak változásokat, és ezáltal a rendszer állapotában (ld. **Hiba! A hivatkozási forrás nem található.**). Ezt matematikailag a következőképpen fogalmazhatjuk:
- Vannak olyan módszerek, ahol megengedik, hogy adatelem 'keletkezzék', és ha ezt kifejezetten specifikálják, akkor vannak CASE eszközök is, amelyek még konzisztencia ellenőrzést is támogatják. Adatelem, jelentések készítésénél, lekérdezéseknél keletkezhetnek; úgy nevezett leszármaztatott elemek jöhetnek létre, más elemek összege, átlaga, stb.

3.1.1.3 FOLYAMATOK

Egy folyamat az adatokat módosítja, manipulálja a rendszeren belül. Általában a szervezet valamely tevékenységét jelenti, amelynek lefolyását valamilyen adat megérkezése kezdeményezi, az adatok aktualizálása után valamilyen kimenetet készít és küld tovább. Nem szabad összekeverni a számítógép programokkal! Előfordulhat, hogy bizonyos esetekben egy az egyben megfelel egy folyamat egy programnak, de ekkor is a felhasználók számára érthető kifejezéseket kell használni a leírására és nem informatikai szakkifejezéseket. Vagyis ezen a ponton szervezet működését kell visszatükrözni. Ez lehet valamilyen számítás elvégzése, új dokumentum készítése azokból az adatokból, amelyek elindították, vagy a bejövő adatokat módosítja.

Az adatfolyam ábra tehát olyan folyamatokat mutat, amelyek az adatok átalakításával foglalkoznak és nem azokat, amelyek csak a jelentésekhez szükséges formátumokat alakítják ki. Ez alól a szabály alól csak akkor lehet kivételt tenni a Jelenlegi Fizikai Adatfolyam ábrán, amikor a felhasználó szemében a jelentés készítés a rendszer nagyon lényeges részének számít.

A folyamatot egy téglalappal jelenítjük meg. A hivatkozási szám a kisebb téglalapban csupán egy azonosító, nem fejez ki sem sorrendet, sem fontosságot. A hosszabb sávszerű téglalapban arra a helyre, helyszínre lehet hivatkozni a Fizikai Adatfolyam ábrán, ahol a folyamat végbemegy. Ahogy az adatfolyam ábra egyre pontosabbá válik úgy lehet egyre pontosabbá tenni a helyszínre való hivatkozást is, ez egyébként lehet annak az alkalmazottnak a megnevezése is, aki végrehajtja ezt a tevékenységet. A folyamatnak egy tömör, világos nevet kell adni. Az egyik lehetőség az, hogy egy igei kifejezést, - felszólító módban - és főnévi kifejezést alkalmazunk. Az ige a tevékenységet írta le, a főnév pedig azt a tárgyat, amelyet a folyamat az igével

kifejezett cselekvéssel manipulál. Ez a megoldás magyarul sokaknak furcsának tűnik, ezért úgy módosítható, hogy az igei kifejezés - *ás*, - *és* képzővel főnevesített változatát használhatjuk. Elemzési szempontból azonban nagyon fontos, hogy a folyamat *aktív (ige)* és *passzív (főnév)* része egyértelműen megjelenjen. (17. ábra).

3.1.1.4 ADATTÁROLÓ

Az adattároló az a hely, ahol az adatok nyugalomba jutnak, ez a hely az - ahogy azt a neve is mutatja -, ahol egy ideig a rendszeren belül az adatokat tároljuk. Ez lehet egy iratszekrény, iktatókönyv, indexelt kartonok, dossziék, elintézendő ügyiratok tálcája, főkönyv, vagy egy számítógépes adatállomány. Az SSADM jelölés egy egyik rövidebb oldalán nyílt téglalapot használ, a baloldalán egy négyzettel, ami tartalmaz egy betűt és egy számot azonosítónak⁸:

- **D**: számítógépes adatállomány;
- **M**: manuális adattároló, pl. iratszekrény, napló, stb.;
- **T(M)**: manuális adattároló. Ez egy ideiglenes adattároló, ami azt jelenti, hogy addig tartják az adatot (adatrekordot) itt, amíg *egyetlen egyszer* ki nem olvassák, azaz az olvasás *destruktív*, az olvasással együtt a tárolóból eltávolítódik vagy törlődik az adatrekord, pl. az elintézendő ügyiratok tálcája, vagy egy postaláda.
- **T**: számítógépes adattároló. Az előbbi adattároló elektronikus változata., pl. egy csak az adatok sorba rendezése miatt ideiglenesen létrehozott adatállomány.

Az adattároló nevének a benne tárolt adatok tartalmát kell kifejeznie, nem pedig a tárolás módját, tehát pl. 'Iratszekrény' helyett 'Személyzeti nyilvántartás'-t kell használni.

Az adattárolóba kerülő és tárolt adatoknak (adatrekordoknak) egyértelműen, egyedileg azonosíthatóknak kell lenniük. Maguk az adatrekordok kezdetben (Jelenlegi Fizikai DFD) lehet, hogy nem optimális szerkezetűek (nincsenek 3. normál forma alakban, 3. NF), de egy úgy nevezett kulcsuknak lenniük kell.

3.1.1.5 AZ ADATFOLYAM DIAGRAM (DFD) KÉSZÍTÉSÉNEK LÉPÉSEI

Az alap fogalmak és jelölések megismerése után a diagram készítésének módszerével ismerkedünk meg részletesebben. Az eddigiek alapján az adatfolyam diagram (DFD) legfontosabb céljait a következőkben foglalhatjuk össze:

- hogyan lépnek az információk be a rendszerbe és hogyan hagyják el;
- mi változtatja meg az információkat;
- miben és hol tárolják az információkat.

Az adatfolyam diagrammot a rendszerelemzésben mint eszközt a következő célokra használjuk:

- *A rendszer határának megállapítása.* A diagram világosan megmutatja a rendszer határait és kiterjedését; a külső entitások és a rendszer határát átlépő adatfolyamok mutatják ezt meg explicit módon.

⁸ Alternatív jelöléseket lsd. 16. ábra.

- *Az elemzés teljességének leellenőrzése.* A diagram készítésének módja, illetve összevetése más modellekkel, amelyek más nézőpontból készültek, biztosítják, hogy az összes információ-folyamat, információtároló helyet, és a rendszer összes tevékenységét feltárták.
- *További specifikációk alapja.* A módszertantól függően szolgálhat a funkciók specifikációjának, vagy logikai rendszertervnek, esetleg programtervnek az alapjául.

SSADM-ben a következő adatfolyam diagramok jöhetnek elő a fejlesztés során:

- *Jelenlegi Fizikai.* A jelenleg működő rendszert modellezi a pillanatnyi állapotában, megvalósításában.
- *Logikai.* Tisztán logikai ábrázolása a jelenlegi rendszernek, amelyet a *Jelenlegi Fizikai* adatfolyam diagramból a fölösleges, fizikai korlátok megszüntetésével vezettek le.
- *Rendszerszervezési alternatívák.* Több alternatív rendszerszervezési tervet készítenek, amelyek mindegyike kielégíti a legfontosabb rendszerrel szemben támasztott követelményeket. Ezeket az alternatívákat áttekintő szintű adatfolyam diagrammokon ábrázoljuk.

4. Módszertanok

A rendszerelemzés legfontosabb technikáinak, valamint néhány tervezési eljárás rövid ismertetése után a *legelterjedtebb* módszertanokról adunk egy összefoglaló képet, tekintettel az osztályozási szempontjainkra (**Hiba! A hivatkozási forrás nem található.**). A világon több száz különböző módszertant használnak, egy felmérés 300-ra teszi a számukat. De valószínűleg még ennél is nagyobb a többé-kevésbé hasonló módszertanok száma, amelyeket valahol leírtak, publikáltak, vagy egyes cégek egy bizonyos változatot kötelezővé tettek belső használatra. Vannak olyan vélemények is, hogy a különbség az egyes módszertanok között valójában triviális, a hangsúlyozott különbségek csak a piaci versenyben játszanak szerepet.

4.1 Strukturált módszertanok

A strukturált módszertanok fejlődése több mint két évtizedre tekint vissza. Ennek köszönhetően a legjobban kidolgozott, érett módszertanok, alkalmas eljárás és technika halmazzal tartoznak ide. Ezek a módszertanok azért jöttek létre, hogy az információrendszer-tervezés minőségi problémáit kezeljék illetve enyhítsék (2.2). Ezt abban próbálták megfogalmazni, hogy a leendő információrendszer "használatossága" jó legyen; ami persze ebben formában meglehetősen homályosan hangzik.

4.1.1 Jobb végeredmény

A következőkben felsorolunk egy kritérium rendszert, amely egy kicsit közelebbről körül járja az információrendszer "használatosságának" fogalmát:

- *Elfogadhatóság:* azok számára, akik a rendszert használják, vajon a rendszer megfelelő-e, kielégíti-e az általuk támasztott követelményeket.

- *Rendelkezésre állás:* vajon a rendszer rendelkezésre áll-e ott és akkor, amikor arra szükségre van.
- *Kohézió:* a rendszer alkotórészei (a részrendszerek) közötti információ csere vajon olyan-e, hogy a végeredmény egy összehangolt működésű, egységes (integrált) információrendszer.
- *Kompatibilitás:* vajon bármelyik adott részrendszer zavartalanul illeszkedik-e be a az egész integrált rendszerbe.
- *Dokumentáció:* vajon létezik-e olyan jó minőségű dokumentáció, amely segíti a felhasználók, a fejlesztők, a vezetők és az üzemeltetők közötti kommunikációt.
- *Megtanulható:* új felhasználók esetében a betanulási görbe vajon elég rövid-e.
- *Gazdaságosság:* vajon maga az információrendszer eredményez-e költségtakarékosságot, és a tervezett erőforrás felhasználáson és egyéb korlátokon belül marad-e.
- *Hatékonyaság:* vajon a rendszer hatékonyan használja fel és ki a rendelkezésre álló erőforrásokat.
- *A fejlesztés gyorsasága:* vajon az információrendszer készítésére fordított idő viszonylag rövid volt-e a rendszer méretéhez és bonyolultságához képest.
- *Rugalmasság:* vajon könnyű-e a rendszert módosítani, új alkotórészt beépíteni vagy egy régit eltávolítani.
- *Funkcionalitás:* vajon a rendszer gondoskodik-e a rendszer (funkcionális) követelményeinek megvalósításáról.
- *Alkalmazásba-vehetőség:* vajon a régi rendszerről az új rendszerre az áttérés megvalósítható-e.
- *Gyenge csatolás:* a részrendszerek közötti kommunikáció, információcsere vajon olyan-e, hogy a részrendszerek úgy módosíthatók, hogy annak nincs hatása a rendszer többi részére.
- *Karbantarthatóság:* vajon mekkora erőfeszítésre van ahhoz szükség, hogy a rendszer üzemeltetését folyamatosan fenntartsák.
- *Hordozhatóság:* vajon az információrendszer tud-e más környezetben, hardver konfiguráción, platformon, más osztályon futni.
- *Megbízhatóság:* vajon a hibák előfordulásának gyakorisága alacsony-e, a kimenetek helyesek-e, és ellentmondásmentesek-e.
- *Robosztusság:* vajon a rendszer hibatűrő-e és hiba esetén visszaállítható-e valamilyen alapállapotból, vagy esetleg automatikusan saját maga is megteszi ezt.
- *Biztonság:* vajon az információrendszer elég robusztus-e ahhoz, hogy szándékos vagy véletlen helytelen használat ne vezessen a rendszer teljes összeomlásához

- *Egyszerűség*: vajon a kétértelműségeket, redundanciákat, bonyolult megoldásokat minimalizálták-e.
- *Teszteltség*: vajon a rendszert alaposan tesztelték-e azért, hogy elkerüljék az üzemelési hibákat és a felhasználók elégedetlenségét.
- *Reszponzivitás*: vajon a rendszer normálisan működik-e a csúcs, normális és a minden egyéb feltételű terhelések időszakában, és akkor és ott szolgáltatja az információt, amikor és ahol az szükséges.
- *Használhatóság*: a használat egyszerűsége, könnyűsége, funkcionalitása, stb., amely megfelel a különböző típusú felhasználók igényeinek (alkalmi, rendszeres, tapasztalt).
- *Átláthatóság*: vajon a felhasználó számára nyomon követhető-e, hogy bizonyos tevékenységek miért bukkantak fel, és miért úgy hajtották végre.

Természetesen optimális egyensúly a fentebb felsorolt kritériumok között nem hozható létre, hiszen vannak olyanok, amelyek csak egymás rovására elégíthetők csak ki. A módszertan illetve a projekt maga hangolható úgy, hogy azok a szempontok kapjanak hangsúlyt, amelyek különösen fontosak az adott probléma területén.

4.1.2 SSADM

Az **SSADM** (Structured Systems Analysis and Design Method) tulajdonosa a CCTA (Central Computer and Telecommunications Agency), amely Nagy-Britannia pénzügyminisztériumához tartozik, és a kormányzati információs rendszerek beszerzése és készítése felett lát el felügyeletet, valamint az információs rendszerek és az informatika területén a kormányzati politikát alakítja ki. A továbbfejlesztését a Nemzetközi SSADM Felhasználók Csoportja (International SSADM User's Group, ISUG) illetve egy arra illetékes testülete felügyeli. A Brit Számítógéptudományi Társaságon belül (British Computer Society, BCS) létezik egy olyan testület, amely a szakmai előírások megvalósulását, azok teljesítésének ellenőrzését egy vizsgáztatási rendszer kialakításával

Az **SSADM** tulajdonképpen eljárási, műszaki és dokumentációs szabványok gyűjteménye, amelyet úgy terveztek meg, hogy kifejezetten a rendszerelemzést és a szoftverfejlesztést támogassa. Két főrészből áll, az egyik a felhasználói követelmények elemzése, a másik a rendszer tervezése. Ezeket a részeket szakaszokra és lépésekre tagolja. A szakaszok összessége lefedi az adatmodellezés technikáit, a követelményelemzést és a szoftver tervezést. Az **SSADM** egyik legfontosabb tulajdonsága, hogy rugalmas, azaz az adott (fejlesztési) körülményekhez igazítható, továbbá az egyik leghatékonyabb módszer, amely olyan szervezetek rendelkezésére áll, amelyeknek egy szabványos rendszerfejlesztési filozófiára és megközelítésre van szükségük.

Az **SSADM** nyílt rendszer. Ez azt jelenti, hogy nyilvános, bárki számára hozzáférhető, bárki használhatja licenc díj fizetése nélkül, engedélyt sem kell kérni a CCTA-tól.

Ez a nyíltrendszer-stratégia egybeesik más egyéb kormányzati, nyílt szabványnál követett eljárással Nagy-Britanniában (pl. OSI, POSIX). Kifejezetten úgy tervezték,

hogy a megjelenése a piacot újra szabályozza és a versenyt a termékek és a szolgáltatások (pl. konzultáció) között fokozza, valamint felszabadítsa a piacot azoktól a korlátoktól, amelyet a tulajdonosnak fizetendő licencdíjak jelentenek. Az **SSADM** stratégia egyik legfontosabb célja, hogy biztosítsa a szolgáltatási piac hatékony működését, a felhasználói igényeket a piaci lehetőségek maximumáig kielégítse. Ily módon a fejlesztésért felelős vezető nem kerül kiszolgáltatott, függő helyzetbe a konzultációt, oktatást és a megvalósítást végző személyektől, ha azokat egy idő után nem találja már a legalkalmasabbnak a feladat ellátására; ilyen esetben a szerződéses partnereket másikkal helyettesítheti anélkül, hogy a befektetések (pénz, idő, stb.) elvesznének.

Az **SSADM** 4.0 és 4.2 verziójában pontos útmutatások találhatók, hogy hol és hogyan alkalmazzák a minőségbiztosítási szabványokat ill. a kapcsolódó eljárásokat, nevezetesen az ISO 9001-t. Ezek az útmutatások nagymértékben rögzítik az ISO 9001 minőségellenőrzési eljárásai bevezetésének a módját azok számára, akik ezt alkalmazni kívánják.

A projekt vezetés/irányítást a **PRINCE** módszertan adja, ami jól összeillik az **SSADM**-mel.

Filozófia:

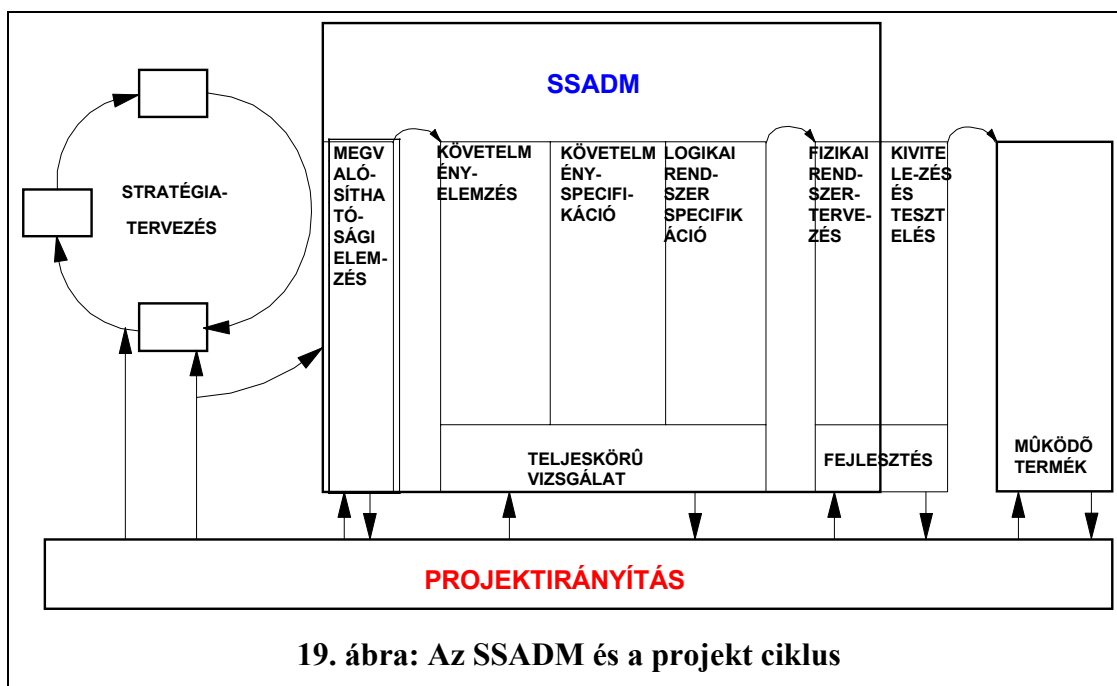
Az **SSADM** alapfilozófiája a különböző nézetek tudatos ütköztetésére, az adatvezérelt, folyamatközpontú és eseményirányított megközelítések tudatos kompromisszumának kialakítására törekszik. Alapvetően felülről-lefelé haladó és adatközpontú elemzési és tervezési módszer, valamint nagy hangsúlyt helyez a felhasználók bevonására. Stuktúrált módszertan és ezért a tudományos alapúak közé soroljuk.

Modellek:

Az entitás-kapcsolat, adatfolyam, entitás-élettörténet és esemény-hatás diagrammok (Jackson-szerű diagrammok) valamint a relációs technika azok legfontosabb eszközök, amelyekkel modellezi, leírja a rendszert.

Életciklus lefedése:

Az SSADM nem foglalkozik informatikai stratégiai tervezéssel - (de feltételezi a létét, pontosabban a rövid projekt specifikációk / meghatározások létét) - az opcionális Megvalósíthatósági tanulmány készítéstől a Fizikai rendszertervezéséig terjedően fedi le a rendszerelemzés és a - tervezés szakaszait. Vagyis csak részleges, nem teljes az életciklus lefedése. Az SSADM nem foglalkozik a rendszerkészítés, karbantartás, üzembe helyezés, és egyéb kiegészítő területek módszereivel ide értve a projektirányítást is.



Leszállítandó termékek:

Alapértelmezésben mindegyik szakasz végén egy pontosan meghatározott dokumentáció készletet kell átadni, amelyeket az adott szakaszban alkalmazott modellezési eljárások és technikák eredményeit tartalmazzák. Például az adatfolyam modell és a logikai adatszerkezet dokumentumait.

Előfeltételezések:

Az SSADM feltételezi, hogy a rendszerfejlesztés célja egy információrendszer létrehozása, azaz egy adatbázis-központú, tranzakció-orientált rendszer elkészítése. Feltételezi, hogy létezik a szervezet meghatározott, kezelhető méretű részére vonatkozó projekt alapító okirat, amire alapozva indulhat a munka. Továbbá a szervezetben van elfogadott projektirányítási módszertan és gyakorlat, valamint több területre kiterjedő szabályok, helyi szabványok és előírások (szervezeti és alkalmazási szintű felhasználói felület tervezési útmutatók, programozási, kódolási, biztonsági szabványok, stb.).

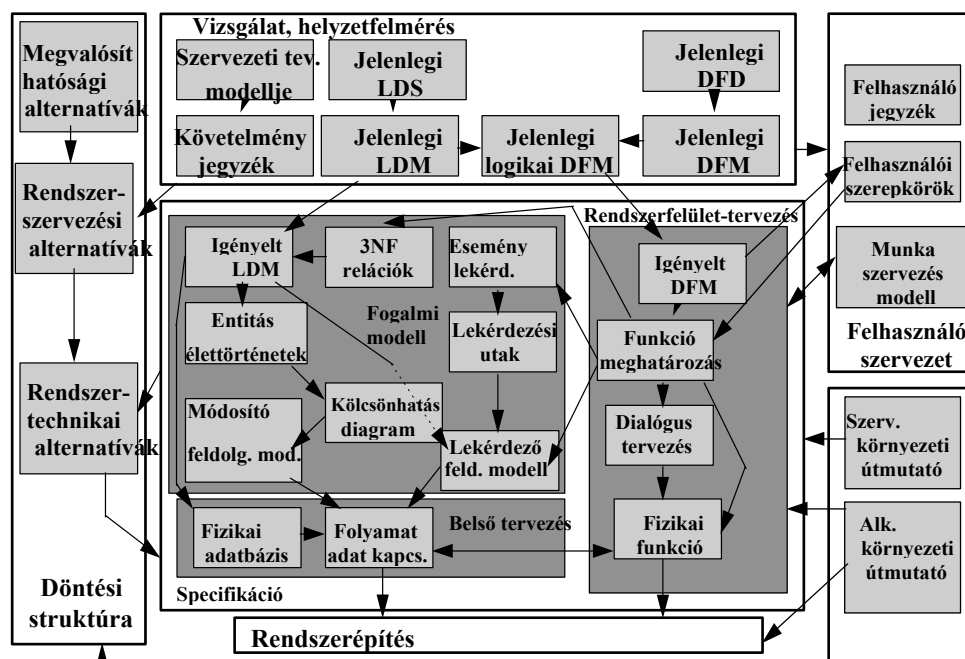
4.1.2.1 DÖNTÉSI PONTOK AZ SSADM-BEN

A hagyományos rendszerfejlesztési eljárásokban a végfelhasználók meglehetősen passzív szerepet játszottak, ők látták el a rendszerelemzőt információkkal és a specifikáció ellenőrzésében valamint a rendszer tesztelésében vettek részt. Azonban semmi esetre sem jöhetett az szóba, hogy befolyásolják vagy megpróbálják befolyásolni a rendszer tervét. Ilyen körülmények között a felhasználó hajlott arra, hogy elfogadja azt a tervet, amit megoldásként adtak neki anélkül, hogy a végfelhasználók kellő időben megkérdőjelezhették volna a terv alkalmasságát. Ennek az eljárásnak aztán számos súlyos következménye támadt.

Az **SSADM** ezzel szemben teljesen eltérő szerepet szán a végfelhasználóknak, ugyanis nekik kell mindazon kritikus döntéseket meghozni, melyek lényegesen

befolyásolják a fejlesztés további menetét. Konkrétan három ilyen fontos döntési pont van:

- **A megvalósíthatósági tanulmány:** A rendszer terjedelme, határa, legfontosabb paraméterei, a rendszerfejlesztés stratégiája a végfelhasználók igényének megfelelően az ő egyetértésükkel kerül meghatározásra.
- **Rendszerszervezési alternatívák:** Lényegében azt határozzák meg, hogy a rendszernek tulajdonképpen MIT is kell csinálnia.



20. ábra A rendszerfejlesztési alapminta és a technikák

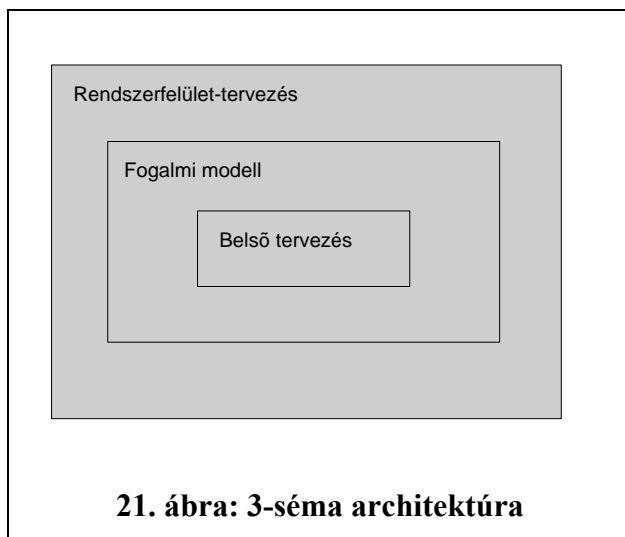
- **Műszaki megvalósítás alternatívái:** Ekkor a kiválasztott rendszerszervezési alternatíva lehetséges technikai/műszaki megoldásai közül választanak a végfelhasználók, ezek a megoldások többnyire széles skálán demonstrálják a szóba jövő műszaki opciókat. A kiválasztás megtörténte után világossá válik, hogy a rendszer HOGYAN fogja megvalósítani azt, amit szolgáltatnia kell.

4.1.2.2 AZ SSADM KULCSFOGALMAINAK HÁTTERE

Az SSADM4+ két kulcsfogalma: a fejlesztési alapminta és a 3-séma architektúra (ld. 20. ábra, 21. ábra). Az SSADM kezdetben a szervezet működésének (üzleti környezetének) a vizsgálatára koncentrált azért, hogy minél jobban meg tudja határozni a leendő rendszerrel szemben támasztott követelményeket. Majd a(z informatikai) rendszerrel leírását, specifikációját és a kapcsoló-felületeket a valóságban működő szervezeti folyamatokhoz határozza meg. Az elemzés és a tervezés termékeit erre a fejlesztési alapmintára lehet leképezni, ezen megtalálhatóak a rendszerfejlesztés legfontosabb területei, nevezetesen:

- helyzetfelmérés;
- specifikáció;

- rendszerkészítés;
- felhasználói környezet;
- döntési pontok;
- szervezeti célok, politikák és eljárások.



21. ábra: 3-séma architektúra

Az információrendszerek készítésekor különbséget teszünk a(z informatikai) *rendszer* és a külvilág között. A *rendszer* specifikáció három fő területét jeleníti meg a 3-séma architektúra, ezen keresztül lehet látni azt, hogy az egyes termékeknek és technikáknak mi a feladata voltaképpen és a módszertan testreszabott verziója készítésekor világossá válik, hogy a séma egyes elemei közül mit és milyen mértékben kíván a testreszabott változat megcélózni és teljesíteni.

- Fogalmi modell:
 - ◆ a szervezeti, működési szabályok;
 - ◆ Logikai Adatmodell;
 - ◆ Entitás Viselkedés Modell;
 - ◆ Fogalmi szintű Adatfeldolgozó Folyamatok Modellje.
 - ◆ Ez a rendszer modell független a felhasználói felülettől, és különböző hardver és szoftver környezetben megvalósítható. A megvalósítás egyik lehetséges módja az, hogy a logikai adatfeldolgozó folyamatokat úgy készítik el, hogy azok a logikai adatmodell entitásain végezzenek olvasási és írási műveleteket.
- Rendszerfelület-tervezés (Külső terv):
 - ◆ felhasználói felület, ember-gép párbeszéd;
 - ◆ be- és kimeneti adatok, állományok;
 - ◆ képernyők, jelentések;
 - ◆ dialógus tervek, programok, kötegetelt adatfeldolgozás be- és kimeneti programjai.
 - ◆ A rendszerfelület terve egy kompromisszum:
 - ♣ a szervezet felépítése;
 - ♣ a rendszer hatékonysága, teljesítménye;
 - ♣ a végfelhasználói felület megvalósításának technológiája;

Összefoglalás

- ♣ az egyes felhasználók egyedi kívánságai;
 - ♣ a biztonsági, auditálási előírások, stb.
- ◆ között.
- A Belsőterv:
 - ◆ fizikai adatterv (esetleg optimalizált a teljesítmény igényekre);
 - ◆ adatfeldolgozó folyamatok és fizikai adatok közötti kapcsoló felület (folyamat-adat kapcsolat);
 - ◆ A fizikai terv is egy kompromisszum:
 - ♣ a válaszidők, időzítési és idő korlátok;
 - ♣ háttértár;
 - ♣ karbantarthatóság;
 - ◆ között.

4.2 Objektum-orientált módszertanok

UML / OMT – Unified modeling Language / Object Modelling Technique (Objektum Modellezés Technikája)

Ezt a módszertan családot is érdemes a teljesség kedvéért megemlíteni. Az UML az OMG (Object Management Group) felkérésére készült el. A cél az volt, hogy egy egységes, szabványos jelöléstechnikát alakítsanak ki az objektum-orientált alkalmazások leírására. Az UML a technológia jelenlegi állásának megfelelően, a legkorszerűbb jelöléstechnikát tartalmazza. A folyamatos fejlődés eredményeként több módszer és jelöléstechnika eredményeit hasznosítja (Booch féle, Object Modeling Technique, OOSE, Object Oriented Software Engineering, OMT).

Vannak akik azt állítják, hogy ez a módszertan lesz a legelterjedtebb típus ([Rumbaugh91], [Shlaer88], [Meyer88], [Coad91]). Ez a megközelítési mód bizonyos területeken nagy sikereket ért el; nevezetesen a termékként forgalmazott szoftverek körében, a felhasználói és grafikus felületek, ember-gép kapcsolat területén (pl. ablakos rendszerek). Ez a megközelítési mód terjedőben van az információrendszerek területén is, de a terjedés sebességét korlátozza az alaptechnológia nevezetesen az információrendszerek készítésére alkalmas objektum-orientált fejlesztő rendszerek — objektum-orientált adatbáziskezelő rendszerek — elterjedtségének hiánya.

5. Összefoglalás

Ebben a jegyzetben meg próbáltuk a legfontosabb rendszerelemzési - tervezési módszereket, alaptechnikákat ismertetni, amelyek ismerete a szakma alap műveltségéhez tartozik. Néhány elterjedt módszertant mutattunk be röviden, a legfontosabb tulajdonságaikat kiemelve.

A jegyzet olvasója egy átfogó képet a módszertanokról és a hozzájuk kapcsolódó legelterjedtebb fogalmakról, ezáltal könnyebb helyzetben a különböző módszertanok

hasonlóságának és különbségének felbecslésére, a további tanulmányozásra érdemes módszerek, esetleges alkalmazások kiválasztására.

6. Bibliográfia

6.1 Idegen nyelvű

- [Ashworth93] Asworth, C., Slater, L., *An Introduction to SSADM Version 4*, McGraw-Hill Book Company, London, 1993. ISBN 0-07-707725-3
- [Bertalanffy68] Von Bertalanffy L., *General System Theory: Foundations, Development, Applications*, Penguin, London, 1968.
- [Booch91] Booch, G., *Object-Oriented Design*, Benjamin/Cummings, Redwood City, Calif., 1991.
- [Burgess90] Burgess, R. S., *Structured Program Design: using JSP*, IEEE Stanley Thorners (Publishers) Ltd., Cheltenham, 1990. ISBN 0 7487 0360 8
- [Cameron83] Cameron, J.R., *JSP and JSD: The Jackson Approach to Software Development*, IEEE Computer. Soc., 1983.
- [CCTA90] CCTA (Central Computer and Telecommunication Agency), *SSADM Version 4 Reference Manuals*, Vols 1,2,3,4 NCC Blackwell, Manchester, Oxford, 1990.
- [CCTA91] CCTA (Central Computer and Telecommunication Agency), *PRINCE Structured Project Management*, NCC Blackwell Ltd., Manchester, Oxford, 1991.
- [CCTA95] CCTA (Central Computer and Telecommunication Agency), *SSADM Version 4+ Reference Manual*, Vols 1,2,3 NCC Blackwell, Manchester, Oxford, 1995.
- [CCTA95A] CCTA (Central Computer and Telecommunication Agency), *SSADM Version 4+ Users Guide*, Vols 1,2,3 NCC Blackwell, Manchester, Oxford, 1995.
- [CCTA95B] CCTA (Central Computer and Telecommunication Agency), *Euromethod in Practice*, NCC Blackwell, Manchester, Oxford, 1995.
- [Coad91] Coad, P., Yourdon, E., *Object-Oriented Analysis, Second Edition*, Yourdon Press, Prentice Hall, Englewood Cliffs, New Jersey, 1991. ISBN 0-13-629981-4
- [Checkland81] Checkland, P., *Systems Thinking, Systems Practice*, John Wiley, Chichester, 1981.
- [Checkland90] Checkland, P., Scholes, J., *Soft Systems: Methodology in Action*, John Wiley, Chichester, 1990.
- [Chen76] Chen, P. P., 'The Entity-Relationship Model: Towards a Unified View of Data', *ACM Transactions on Database Systems*, Vol. 1, No. 1, March 1976, pp 9-36, 1976.
- [Chen81] Chen, P. P.-S., 'A Preliminary Framework for Entity-Relationship Models', In P. P.-S. Chen (ed.), *Entity-Relationship Approach to Information Modeling and Analysis*, ER Institute, PO Box 617, Saugus, Calif. 91350, 1981.
- [Churchman68] Churchman, C., W., *The Systems Approach*, Dell Publishing Co., New York, 1968.
- [Date90] Date, C. J., *An Introduction to Database Systems*, Addison-Wesley, 1990.

Bibliográfia

- [Downs92] Downs, E., Clare, P., Coe, I., *Structured Systems Analysis and Design Method, Application and Context*, Second Edition, Prentice Hall International (UK) Ltd., New York, London, 1992.
- [Eva92] Eva, M., *SSADM Version 4: A user's guide*, McGraw-Hill, 1992.
- [Gane79] Gane, C., Sarson, T., *Structured Systems Analysis: Tools and Techniques*, Prentice-Hall, NJ, 1979.
- [Gane90] Gane, C., *Computer Aided Software Engineering, the methodologies, the products and the future*, Prentice-Hall, 1990.
- [Hares90] Hares, J. S., *SSADM for the Advanced Practitioner*, John Wiley & Sons, Chichester, England, 1990.
- [Hewett89] Hewett, J., Durham, T., *CASE: The next step*, Ovum Ltd., 1989.
- [Hußmann94] Hußmann, H., *Formal Foundations for SSADM, An approach Integrating the Formal and Pragmatic Worlds of Requirements Engineering*, FAST-Bericht Nr. 94-02, Forschungsinstitut für Angewandte Software-Technologie e. V. (Arabellastr. 17, D-8195 München, Germany, e-mail:info@fast.de), Juni 1994.
- [Jackson82] Jackson, M., A., *System Development*, Prentice Hall, Englewood Cliffs, 1982. ISBN 0-13-880328-5
- [Griethuysen82] van Griethuysen (ed.), *Concepts and terminology for the conceptual schema and the information base, computers and information processing*, ISO/TC97/SC5/WG3 International Organisation for Standardisation, Geneva, Switzerland, 1982.
- [Longworth86] Longworth, G., Nichols, D. *SSADM Manual Vol. 1-2*, NCC Blackwell, 1986.
- [Longworth88] Longworth, G., Nichols, D., Abbot, J., *SSADM Developer's Handbook*, NCC Publications, Blackwell, Manchester, 1988.
- [Layzell89] Layzell, P., Loucopoulos, P., *Systems Analysis and Development*, Chartwell-Bratt, Studentlitteratur, 3rd edition, 1989. ISBN 0-86238-215-7.
- [Martin81] Martin, J., *Design and strategy for distributed data processing*, Prentice-Hall, Englewood Cliffs, New Jersey, 1981. ISBN 0-13-201657-5
- [MartinFin81] Martin, J., Finkelstein, C., *Information Engineering*, Vols. 1. and 2., Prentice Hall, Englewood Cliffs, New Jersey, 1981.
- [Martin89] Martin, J., *Information Engineering: a trilogy*, Vol. 1, *Introduction*, Prentice Hall, Englewood Cliffs, New Jersey, 1989. ISBN 0-13-464462-X
- [Martin88] Martin, J., McClure, C., *Structured Techniques, The Base for CASE*, Prentice Hall, Englewood Cliffs, New Jersey, ISBN 0-13-854936-2.
- [Matheron90] Matheron, J. P., *Comprendre Merise, Outils Conceptuels et Organisationnels*, Editions EYROLLES, 1990.
- [Meyer88] Meyer, B., *Object-Oriented Software Construction*, Prentice Hall, Hertfordshire, England, 1988.
- [Olle91] Olle, T. W., Hagelstein, J., Macdonald, I. G., Rolland, C., Sol, H. G., Van Assche, F. J. M., Verrijn-Stuart, A. A., *Information Systems Methodologies: A framework for understanding*, 2nd. edition, Addison-Wesley, Wokingham, England, 1991.
- [Pham91] Pham Thu Quang, Chartier-Kastler, C., *MERISE in Practice*, Macmillan Education Ltd, Houndmills, 1991.
- [Polack94] Polack, F., Whiston, M., Mander, K. C., *The SAZ method, version 1.1*, University of York, January 1994.

- [Rochfeld83] Rochfeld, A., Tardieu, H., 'Merise: An information system design and development methodology', *Information Management*, Vol. 6, No. 3., pp. 143-159, 1983.
- [Rumbaugh91] Rumbaugh, J., Blaha, M., Premerlani, W., Eddy, F., Lorensen, W., *Object-Oriented Modeling and Design*, Prentice Hall, Englewood Cliffs, New Jersey, 1991. ISBN 0-13-630054-5.
- [Shlaer88] Shlaer, S., Mellor, S. J., *Object-Oriented Systems Analysis: Modeling the World in Data*, Yourdon Press, Englewood Cliffs, New Jersey, 1988.
- [Skidmore92] Skidmore, S., Farmer, R., Mills, G., *SSADM Models and Methods Version 4*, NCC Blackwell, Manchester, Oxford, 1992. ISBN 0-85012-796-3
- [Turner90] Turner, W. S., Langenhorst, R. P., Hice, G. F., Eilers, H. B., Uijttenbroek, A. A., *SDM system development methodology*, Elsevier Science Publishers B.V. (North-Holland)/Pandata, 1990.
- [Turner96] Turner, P., Jenkins, T., *Euromethod and Beyond, Open Frameworks for European Information Systems*, International Thomson Computer Press, London, 1996.
- [Ungoed94] Ungoed, A., *A model of SSADM concepts for non-SSADMers*, Thesis, Brighton University, U. K., 1994.
- [Ward85] Ward, P. T., Mellor, S. J., *Structured Development for real-time systems*, Volumes I-III. New York, Yourdon Press, 1985-86.
- [Warnier81] Warnier, J. D., *Logical Construction of Systems*, Van Nostrand Reinhold, New York, 1981.
- [Yourdon75] Yourdon, E., Constantine, L., *Structured Design*, Yourdon Inc., New York, 1975.
- [Yourdon89] Yourdon, E., *Modern Structured Analysis*, Prentice-Hall International, Inc., Englewood Cliffs, NJ, 1989.

6.2 Magyar nyelvű

- [Bana94] Bana István: *Az SSADM rendszerszervezési módszertan*, LSI, Számalk, Budapest, 1994.
- [BKE] Budapesti Közgazdaságtudományi Egyetem jegyzet: *Rendszerfejlesztés módszertana, SSADM*, összeállította: Molnár Bálint, adjunktus
- [Halassy94] Halassy Béla: *Az adatbázis tervezés alapjai és titkai*, IDG kft., Budapest, 1994.
- [Kovács95] Dr. Kovácsné Cohner Judit, Takács Tibor: *Ismerkedés az SSADM-mel*, Computer Books, Budapest, 1995.
- [Euromethod94] Euromethod áttekintés,
Euromethod vevői útmutató,
Euromethod szállítói útmutató,
Euromethod kivitelezés tervezési útmutató,
Euromethod esettanulmány, 1994--- Miniszterelnöki Hivatal Informatikai Koordinációs Iroda, Hírlapkiadó.
- [Molnár1997] Molnár Bálint, 'Bevezetés a rendszerelemzésbe', in: Gábor András (szerk.) „Információmenedzsment”, Aula Kiadó, 1997, pp 107-239.
- [Molnár1996-1] Molnár Bálint, Rendszerelemzés, in: Gábor András (szerk.) „Információmenedzsment”, Aula Kiadó, CD melléklet, 1996-98, <http://www.mtaita.hu/Ssadm1.pdf>

Tárgymutató

- [Molnár1996-2] Molnár Bálint, Rendszertervezés, in: Gábor András (szerk.) „Információmenedzsment”, Aula Kiadó, CD melléklet, 1996-98, <http://www.mta.hu/Ssadm2.pdf>
- [Molnár2002] Dr Molnár Bálint: Bevezetés a rendszerelemzésbe, A rendszerszervezés alapjai, Műszaki Könyvkiadó, 2002. (Kiadói azonosító: MK-00275), <http://www.muszakikiado.hu/details.php?details=MK-00275>, <http://www.mta.hu/Bevezetes.pdf>
- [Quittner93] Quittner Pál, *Adatbáziskezelés a gyakorlatban*, Akadémiai Kiadó, Budapest, 1993. ISBN 963 05 6587 0.
- [Vecsenyi88] Vecsenyi János, *Szervezeti problémamegoldás*, Marx Kátoly Közgazdaságtudományi Egyetem, Közgazdasági Továbbképző Intézet, Budapest, 1988.

7. Tárgymutató

A,Á

| | |
|---------------------------------------|----|
| adatfolyam modell | 30 |
| adatmodell | |
| logikai adatmodell | 41 |
| adatszerkezet | |
| logikai adatszerkezet..... | 39 |
| adatszótár | 29 |
| alternatívák | |
| megvalósíthatósági tanulmány..... | 40 |
| műszaki megvalósítás | 40 |
| rendszerszervezési alternatívák | 40 |

C

| | |
|---------------------|--------|
| CASE eszközök | 15, 30 |
|---------------------|--------|

D

| | |
|--------------------|---------------------|
| dialógus | lsd. terv, dialógus |
| döntési pont | 41 |

E,É

| | |
|--------------------------|----|
| esemény | 10 |
| szervezeti esemény | 17 |
| Euromethod | 27 |

F

| | |
|---------------------|----|
| fejlesztés | |
| CASE | 23 |
| evolúciós | 25 |
| inkrementális | 25 |

fejlesztési

| | |
|--------------------------|------------|
| szakasz | 22 |
| fogalmi modell | 41 |
| folyamat | 33 |
| folyamat modell | |
| adatfolyam modell | 30 |
| folyamat modell..... | 12, 13, 14 |
| folyamat modell..... | 29 |
| folyamat modell | |
| adatfolyam ábra | 29 |
| szervezeti folyamat..... | 9, 40 |
| folyamat modell | |
| adatfolyam modell..... | 39 |

I,Í

| | |
|---------------------------|--------------|
| információrendszer | |
| adaptáció | |
| IR-adaptáció | 27 |
| információrendszer | 5, 9, 11, 12 |
| ábrázolásai..... | 12 |
| fejlesztési elvek | 16 |
| modelljei..... | 12 |
| szoftver fejlesztés | 17 |
| információrendszer | |
| működési terület | 28 |
| információrendszer | |
| használhatósága..... | 36 |

| | | | |
|--------------------------------------|------------|--------------------------------------|--------|
| információrendszer | | SSADM..... | 38 |
| fejlesztés | 37 | működési terület..... | 28 |
| információrendszer | | N | |
| karbantarthatóság..... | 37 | negyedik generációs nyelv | 14 |
| információrendszer | | P | |
| hordozhatóság..... | 37 | projekt | 25 |
| információrendszer | | prototípus | 23, 25 |
| megbízhatóság | 37 | prototípus készítés..... | 25 |
| információrendszer | | R | |
| biztonság..... | 37 | rendszer | |
| információrendszer | | adatközpontú megközelítés | 8 |
| reszponzivitás | 37 | információrendszer..... | 2 |
| K | | rendszerelmélet | 4 |
| követelmény | | részrendszer..... | 3 |
| funkcionális követelmény..... | 15 | rendszerek | |
| M | | osztályozása..... | 2 |
| megvalósítás | | rendszerszervezési alternatívák..... | 35 |
| ábrázolása | 13 | rendszerterv | |
| modell | | logikai rendszerterv | 34 |
| fogalmi..... | 25 | Sz | |
| modell | | számítógéprendszer..... | 10 |
| folyamat modell..... | 12, 13, 14 | T | |
| szervezeti | 14 | termék | |
| modell | | átadandó termék | 27 |
| adatfolyam modell | 30 | tervezési termék | 28 |
| modell | | terv | 40 |
| fogalmi modell..... | 41 | dialógus | 41 |
| módszer | 1 | fizikai adatterv..... | 42 |
| módszertan..... | 1, 37, 39 | fizikai terv | 42 |
| Objektum Modellezés Technikája | 42 | rendszerfelület..... | 41 |
| PRINCE | 38 | tranzakció..... | 10 |
| SSADM | | | |
| nyíltrendszer..... | 38 | | |