

**Nagyméretű informatikai beruházásoknál
(fejlesztéseknél) ajánlott szoftveroldali
tervdokumentációk tartalmi elemeinek
meghatározása**

I. kötet



**Magyar Mérnöki Kamara
Kiadványsorozata 2.**

**Nagyméretű informatikai beruházásoknál
(fejlesztéseknél) ajánlott szoftveroldali
tervdokumentációk tartalmi elemeinek meghatározása**

**MMK FAP azonosító:
2018/020-HIT**

Budapest, 2018. október

A sorozat szerkesztője:
NAGY GYULA
a Magyar Mérnöki Kamara elnöke

Készült a Magyar Mérnöki Kamara Hírközlési és Informatikai Tagozatának
gondozásában,
a 2017. évi Feladat Alapú Pályázatok pénzügyi keretéből.

A kiadvány a Magyar Mérnöki Kamara tulajdona. Másolása, teljes terjedelmében való
közzététele csak a Kamara engedélyével lehetséges. Minden jog fenntartva.

Szerzők:

Gábori László Dr.
Beinschróth József Dr.
Nógrádi Gábor
Rátkay Tamás

Lektorálta:

Kakuk Ilona

Kiadó:

Magyar Mérnöki Kamara
1094 Budapest, Angyal u. 1-3.
info@mmk.hu, www.mmk.hu

ISSN: **xxxx-xxxx**

TARTALOMJEGYZÉK

1.	Bevezetés	7
2.	Probléma felvetése	9
2.1.	Probléma leírása.....	10
2.2.	Az informatika illeszkedése a műszaki szakágakhoz	11
2.2.1.	A mérhetőség szerepe.....	11
2.2.2.	Modellezés, tervezés.....	12
2.2.3.	Élettartam, korszerűség	12
2.3.	Célkitűzés	13
3.	Szakmai minőségi célok az informatikai projektekben	15
3.1.	Szemlélet, megközelítés.....	16
3.1.1.	Információ technológiai sajátosságok.....	17
3.1.1.1.	Működési logika, mint eredmény termék.....	17
3.1.1.2.	Tervezés.....	18
3.1.1.3.	Munkamódszerek.....	18
3.1.1.3.1.	Rendszerépítés	18
3.1.1.3.2.	Munkaszervezés	20
3.1.1.4.	Ráfordítás meghatározás.....	20
3.1.1.5.	Egyéb jellemzők.....	22
3.1.2.	Technológiai modellek	23
3.1.3.	Szabványosítás és mindennapi gyakorlat.....	23
3.1.3.1.	Az informatikai rendszerek reprezentációja	24
3.2.	Az információs rendszerek tervezésének és fejlesztésének módszertana	27
3.2.1.	Információs rendszerek tervezési fázisai.....	27
3.2.1.1.	Követelmény kezelés.....	28
3.2.1.1.1.	Követelmény specifikáció.....	29
3.2.1.1.2.	Igény specifikáció	30
3.2.1.1.3.	Korszerű megjelenítés	32
3.2.1.1.4.	Esettanulmány modell.....	34
3.2.1.1.5.	Állapotgép	38
3.2.1.2.	Elemzés.....	39
3.2.1.3.	Architektúra kialakítás.....	41
3.2.1.4.	A tervdokumentáció elvárt tartalmi egységei.....	42
3.2.1.4.1.	A koncepcionális tervezési szint	42

3.2.1.4.2.	A logikai tervezési szint.....	45
3.2.1.4.3.	A fizikai tervezési szint.....	48
3.2.1.5.	Tervezés támogatása.....	52
3.2.1.6.	Szemléltetés	54
3.2.2.	Fejlesztés	58
3.2.2.1.	Fejlesztés támogatása	58
3.2.3.	Dokumentálás.....	59
3.3.	Minőségi célok elérése.....	59
3.3.1.	A tervek értelmezhetősége, a rendszer közérthetősége	60
3.3.1.1.	Szempont.....	60
3.3.1.2.	Módszertan, feladat	61
3.3.2.	A tervezés követhetősége	62
3.3.2.1.	Szempont.....	62
3.3.2.2.	Módszertan, feladat	62
3.3.3.	A tervezési elvek következetessége	63
3.3.3.1.	Szempont.....	63
3.3.3.2.	Módszertan, feladat	63
3.3.4.	Korszerűség, megbízhatóság, nívó	63
3.3.4.1.	Szempont.....	63
3.3.4.2.	Módszertan, feladat	64

1. Bevezetés

Feladat bemutatása

Az informatikai tervező mérnök (Enterprise Architect) feladata a módszeres tervezés gyakorlati alkalmazása, az architekturális kontroll megvalósítása az informatikai beruházási és üzemeltetési folyamatokban.

A digitális világ terjedése (a negyedik ipari forradalom), a mind szerte ágazóbb fejlesztések komoly kihívást jelentenek az infókommunikáció területén dolgozó szakembereknek, mert újszerű projektmenedzsment szemléletre, a megrendelőt képviselő projekt-ellenőri tudásra is szükségük van, s fel kell készülniük a folyamatosan növekvő, szerteágazó IT biztonsági kihívásokra. A szakma újszerűsége, rövid néhány évtizedes múltja magyarázza a szakirodalom esetlegességét, szórtóságát, a szakmai tevékenység alulszabályozottságát, ugyanakkor már jelentkeznek, fellelhetők e hiánynak a negatív jelenségei. Szakembereink szervezett szakmai támogatása nélkül prognosztizálható a súlyosabb káros hatások, következmények bekövetkezése: az informatikára is egyre inkább jellemző lesz az, ami a többi, klasszikus mérnöki szakágnál természetes, hogy életek múlhatnak a hibás tervezésen vagy a gondatlan kivitelezésen.

A tagozat középtávú feladatának tekinti e dinamikusban fejlődő terület szabályozásának előkészítését, javaslatok készítését az elérhető referencia architektúrák hasznosítására, a legjobb gyakorlatok átvételére, honosítására. Továbbá a tagozat támogatja a tervezés és ellenőrzés módszertanának kidolgozását, továbbfejlesztését, az oktatás különböző szintjein történő elterjedését.

A tagozat célja az informatikai beruházások kockázatainak csökkentése, a hardver és szoftver elemek együttműködésében jobban menedzselhető, időállóbb rendszerek létrehozása, melyek módosítása, fejlesztése megbízhatóbban és gazdaságosabban elvégezhető. Az informatikai tervező mérnököknek, a nagyvállalatok mellett kiemelt működési területe az állami fejlesztések, ahol el kell érni, hogy a beszerzett informatikai termékek és szolgáltatások formalizált folyamatokat (protokollokat) tartalmazzanak.

Az elvégzett munka

Munkánk során összeállítottuk a szoftverek fejlesztésével, adaptálásával, telepítésével és üzemeltetésével kapcsolatos műszaki tervdokumentációk tartalmi elemeit, valamint meghatároztuk az IKT (Információs és Kommunikációs Technológia) projektek eredménytermékeivel kapcsolatos tervezési, minőségellenőrzési, audit és üzemeltetési engedélyezési eljárások fő szempontjait. Kiemelten foglalkoztunk az IKT

projektek legproblematisabb területének, a koncepcionális leírásnak a módszertanával.

A kötetek összeállításánál figyelembe vettük a nemzetközi és hazai szakirodalmat, hozzáférhető oktatási, képzési tananyagokat, vállalati belső szabályzatokat. A munkánk során nehézséget jelentett a témakör szerteágazó voltának a kezelése. Ezért javaslatot készítettünk a projektek IKT szempontú tipizálási rendszerére. Az IKT projektek tipizálásának fő dimenziói:

- 1.) informatikai szakterület (témakör)
- 2.) nemzetgazdasági ágazat, gazdasági szektor („szférák”)

Ezen kívül lényeges szempont, hogy a beruházás célja szerint „zöldmezős” vagy egyéb színű a beruházás. (lásd II. kötet)

A dokumentumok összeállításánál még nem vettük figyelembe ennek a „térnek” a szerteágazó voltát, aleteit, hanem egy általános tartalomjegyzék-szerűt állítottunk össze.

További feladatok

A következő lépésben kell majd a szélesebb szakmai közönség segítségével, támogatásával kidolgozni a specialitásokat. Egy ilyen széles körű felmérés alapján választ is kaphatnánk, hogy mi lenne az érdeke és mi lenne az igénye egy mérnöknek, egy szakirányú végzettség nélküli specialistának, mi a megrendelőnek, mi a foglalkoztatóknak, mi az államnak stb.

Várjuk a fenti „tér” bármelyik elemére vagy sorára, oszlopára vonatkozó észrevételeket, javaslatokat, újabb kritériumokat, amik korlátozódhatnak csak a minőségellenőrzésre, de kiterjedhetnek a tervezésre, dokumentálásra, illetve az egész fejlesztési és üzemeltetési folyamat valamennyi elemére is.

A hatékony feldolgozás érdekében kérjük a kollégákat, hogy javaslataikat helyezték el a Zachman besorolási rendszeren alapuló módszertani ábránkban.

Reményeink szerint a feldolgozott javaslatokat fogjuk majd a következő időszakban feldolgozni és megjelentetni.

2. Probléma felvetése

Az informatika a legalapvetőbb infrastruktúrává vált/válik a magán, az állami szektor, az üzleti élet, és a gazdaság minden területén. Az informatika fontosságát mindenki érzi abban a pillanatban, amikor az nem működik: teljesen természetes elvárásunkká vált, hogy minden pillanatban rendelkezésre álljon az üzenetküldés, a mobilparkolás, a közösségi oldalak platformja, vagy a műholdas navigáció az autóban.

Emiatt a fejlesztési projekteknek ugyanolyan fontos része kell, hogy legyen a fenntarthatóság és a kivitelezés utáni üzemeltetés kritériumainak valamint feltételeinek a kialakítása, mint maga a fejlesztési koncepció megalkotása és annak megvalósítása. A nagyméretű, komplex informatikai megoldások egyre inkább olyan környezetben kerülnek kialakításra, ahol a fejlődésnek köszönhetően az eredendően megtervezett és kialakított rendszer kapacitása és paraméterei az idővel változhatnak. Ilyen szintű feladatmegoldáshoz és problémakezeléshez pedig nem feltétlenül a naprakész programozói tudás az alapkövetelmény, mint inkább a rendszerszemléletű gondolkodási képesség, a számos hasonló referencia-rendszerek kialakításában megszerzett tapasztalat, vagyis a mérnöki látásmód és rutin a kulcs. A mérnököktől elvárás, hogy munkájukat a műszaki és engedélyezési szabványok betartásával, az előírt dokumentáltsági szint és igazolt tesztek teljesítésével végezzék. Ettől eltérően sajnos ma gyakorlat, hogy az informatikusok többsége ezeket tipikusan elhanyagolja. Nem követelik meg tőlük, nincs is kultúrája, és magas arányban dolgoznak a szektorban olyan felsőfokú mérnöki és/vagy szakirányú végzettséggel nem rendelkezők, akik rendelkeznek ugyan a szükséges programozói tapasztalattal és tudással, de akiktől a mérnöki látásmód mégsem várható el.

Az infokommunikációs technológia ma már eljutott arra a fejlettségi szintre, ahol nagyon nagy kockázatot jelent a hibás tervezés, a terveknek nem megfelelő kivitelezés, a nem moduláris, nem bővíthető, nem szabványos és nem fenntartható megoldások kialakítása, valamint a nem megfelelő dokumentáltság. A helyzet javítása, normalizálása érdekében a Tagozat középtávú célja, hogy a többi mérnöki szakterülethez hasonlóan, a szakmagyakorlási jogosultságot érvényesítése első körben a nagyméretű informatikai fejlesztések tervezését, kivitelezését és üzemeltetését végző szakemberek körében, az IKT projektek általunk javasolt tipizálás szerinti szakterületein. Ehhez szükséges, hogy megkülönböztetett figyelemmel, szervező munkával, a szakmával történő konzultációval, és széles körű tájékoztatással kerüljön kialakításra a hírközlési és informatikai területen a jogosultsággal végezhető tevékenységek köre. Ezzel együtt lehetővé kell tenni, hogy a nem mérnöki, de felsőfokú végzettségű informatikusok, programozók,

rendszertervezők is kapjanak hasonló lehetőséget a kamarai tagság – tervezői jogosultság - megszerzésére. ¹

Ugyanakkor nem cél, hogy a Tagozat szűkítse a hazai informatikusok mozgásterét, sokkal inkább az, hogy támogassa őket tanúsítással és képzéssel. Fontos annak felmérése is, hogy a hazai közigazgatás számára láthatóvá, nyilvántartottá váljanak végre azok a jól képzett magyar informatikus szakemberek, akikre az állami- és magánmegrendelők egyaránt támaszkodhatnak.

Ennek a folyamatnak az elindítása érdekében készítettük el ezt a dokumentumot, azért, hogy az informatikában is érvényesüljön az a személyes felelősségvállalás, ami a többi műszaki tevékenységben ma már természetes követelmény.

2.1. Probléma leírása

Egyes nagy informatikai beruházások alacsony műszaki színvonalának oka visszavezethető a projekt előkészítése és kivitelezése során tapasztalható hiányos tervezési és ellenőrzési megoldásokra:

- Nem ismerik, illetve nem tartják be a nemzetközi fejlesztési irányelveket.
- Nem használnak a tervezés és a kivitelezés során egységes kollaborációs eszközöket (így nem is vezetik át a módosításokat).
- Több évtizede nincs rendszertervező képzés (felsőfokú képzésben programkódoló illetve programtervező szakirányok vannak).
- Programkódolási munka dokumentálatlan, illetve ha van, ez alacsony színvonalú:
 - nem készül megfelelő kiviteli terv,
 - nincs orientáló, alkalmas magyarázó megjegyzésekkel (comment) ellátva a kód,
 - nincs hivatkozás az algoritmusok átvételénél (forrás megjelölés hiánya),
 - nem kritérium a kód jó olvashatósága.
- A fejlesztők nem érdekeltek a cserélhető, szabványos interfésszel rendelkező, moduláris megoldások kialakításában.
- Programtesztelés döntően manuális, tesztelő szoftverek alkalmazása nem előírás, nem elterjedt gyakorlat.
- Kritikus rendszerek biztonsági besorolási szempontjai nem egyértelműek.

¹ lásd: <https://mmk.hu/ugyintezes/tanusitas>

- Nincs valódi garancia, még a nemzetközi szoftver-gyártók is „As is” módon adják át a rendszereiket, nincs garancia az előírt funkcionalitásnak megfelelő működés biztosítására, javítócsomagokat biztosítanak saját ütemük, és prioritásuk szerint.

Amennyiben egy szoftver vagy alkalmazás meghibásodik, sok esetben egyszerűbb újra írni, mint a hibát kijavítani. Azok az alapvető paraméterek, amiket a hagyományos mérnöki alkotómunkától megkövetelünk, az informatikából ma még hiányoznak.

Még jó nevű cégek is leszállítanak olykor igénytelenül megírt és hibákkal működő szoftverprogramokat, ha az átvételnek nincsenek meghatározva valós minőségi kritériumai. Ebből eredően újra és újra megírásra kerülnek ugyanolyan alkalmazások.

Az egyoldalú megoldás, a *one-man show* nem csak a fejlesztéseknél, hanem az üzemeltetésnél (hibajavításoknál) is káros gyakorlat, túlzott felhasználói kitettséget okoz.

Informatikai rendszerek tervezési és kivitelezési folyamatai nincsenek úgy kidolgozva, illetve nincs olyan gyakorlat, mint a klasszikus mérnöki tevékenységeknél: nincsenek előírt tervezési ajánlások és szabványok, nincs előírás a vezető tervezőre, nincs olyan kategória, hogy tervező és műszaki ellenőr, tipikusan nem független szakembereket bíznak meg ezekre a feladatokra, hanem mindezeket a tevékenységeket többségében maga a szállító biztosítja.

2.2. Az informatika illeszkedése a műszaki szakágakhoz

Vizsgáljuk meg a műszaki tudományokhoz tartozó informatikát a mérhetőség, tervezhetőség, modellezhetőség, valamint az élettartam szempontjából.

2.2.1. A mérhetőség szerepe

A klasszikus mérnöki területeken tervezéskor, kivitelezéskor és ellenőrzéskor, tehát a teljes megvalósítási folyamat során a megfelelő műszaki paramétereket számolják és mérik. A tervezési és ellenőrzési algoritmusok sok évtized alatt alakultak ki, csiszolódtak. Még ma is újabb és újabb műszaki adattípusok kerülnek be a folyamatba, ezzel is fejlődnek, pontosabbá válnak a kivitelezési eljárások. Ezek a paraméterek közvetve vagy közvetlenül „kézzel fogható” objektumokhoz köthetők, ezzel megkönnyítve a mérési probléma megoldását, megoldhatóságát.

Az informatikára, a „virtuális világra” jellemző a mérések nehézsége, nincsenek elvek, módszerek, illetve, ha van egy-egy részterületen, akkor azok az ismeretek parciálisak,

nem terjedtek el. Mérnöki szempontból még gyerekcipőben jár ez az új tudomány, el kell telni egy kis időnek, hogy reményeink szerint felbukkanjanak e tudományterület klasszifikációját létrehozók, megreformálók. Valódi mérések hiányában addig az eljárási folyamatokat tudjuk szabályozni, és az átláthatóságot lehet javítani – és ez sem kis feladat.

2.2.2. Modellezés, tervezés

Amíg a klasszikus mérnöki területeken modelleznek, ha szükséges, addig az informatikában pilot projekttel lehet a bizonytalanságokat feltárni, felmérni. Senkinek nem jut eszébe egy épületnél pilot címen egy emeletet felépíteni, kipróbálni, és ha beválik, felhúzni a többi 29 emeletet, vagy a kipróbálás alatt megismert probléma esetén újra építeni az alapoktól az egészet.

A sikeres, nagy informatikai projekteket két lépésben készítik el. Először pilot keretében egy egyszerűbb működésű, de teljes funkcionalitású alkalmazással, és szűkebb felhasználói körrel tesztelik az elképzeléseket (a terveket), majd az eredmények alapján, ha kell, módosítják, átdolgozzák a terveket és ezek alapján készítik el a végleges változatot. Mivel a pilot költsége nagyságrenddel kisebb, mint a végleges alkalmazást kidolgozó projektté, ezért könnyebben kivitelezhető a menet közbeni teljes koncepcióváltás is. Pilot típusú „elő” projekt nélkül szinte elképzelhetetlen, hogy egy nagy projektben, menetközben más legyen a végtermék, mint amire a partnerek szerződtek, hiszen nehéz felvállalni, szinte lehetetlen a döntéshozóknak a jogilag rögzített céltól történő drasztikus eltérést, a teljes változtatást.

Ugyanakkor léteznek már ma is olyan modellező megoldások, amelyek segítségével a felhasználói igényeknek megfelelően össze lehet állítani a folyamatokat, amelyek akár a programkódot is legenerálják, de ezek a megoldások ma még kevésbé elterjedtek, igen drágák, nem feltétlenül az optimális megoldást szolgáltatják, illetve a fejlesztők még nem érdekeltek ilyen modellező eszközök használatában.

2.2.3. Élettartam, korszerűség

A műszaki, fizikai berendezések élettartama fizikai jellemzőik mentén, akár szemrevételezéssel, akár megfelelő vizsgálati eszközök alkalmazásával, kellő pontossággal, többségében jól meghatározhatók. Az épületek, berendezések elavulása moduláris (jól elkülöníthető) összeépítésük miatt többnyire egyszerű minőségi cserével megoldhatók, még a megváltozott teljesítmény, illetve igénybevétel miatt keletkező, nagyobb méretű átalakítások esetében is. A korszerűség évtizedes távlatokban biztosítható, ellenőrzése pedig szinte rutinfeladat.

Az informatikai fejlődés üteme, a korszerűség mára túlhaladta a négyzetes ugrási jellemzőt, mind a hardver, mind a szoftver területén, sőt, a változások okozta jelentős globális megújulások 2-3 éves gyakorisággal jelentkeznek. Ezeknek a trendeknek a hatására alapvető felhasználói igényné vált az alacsony ráfordítással bővíthető, és gyors alkalmazkodó képességgel rendelkező informatikai rendszerek létrehozása. A korszerűség az informatikában nehezen tetten érhető jellemző, mivel, ahogy a koncepcióról szóló részekben kifejtjük, sok benne a szubjektív elem, amelyek „mérhetősége” nem magától értetődő. Önmagában a – fizikai, logikai - technológia, bármilyen új is, nem biztosítja egyik fenti tulajdonság megvalósulását sem. A háttérben a működési modellen alapuló felépítés, az architektúra játssza a főszerepet, aminek milyensége, minősége a rendszerbe ágyazottan, az elemek, részek koncepcionális kiválasztásában, összeszerkesztésében, együttműködési módjában ölt testet, így mind a felhasználó, mind a minőség ellenőrzője számára egyaránt kihívást jelentő feladat a felderítése, megértése és elfogadása. A nehézség feloldható az egyes szereplők nézőpontjának megfelelő koncepció-nézet meg-, és bemutatásával. Jelenleg a sokoldalú, vizuális modelleken alapuló bemutató a legelterjedtebb eszköz, a pilot funkcionalitás létrehozása mellett, ami nézetünk szerint csökkentheti az ehhez a különleges, mesterséges (virtuális) világhoz való hozzáférés korlátait.

A megvalósuló rendszer korszerűségét a tervekben megjelenő gondolkodási mód, az ebből következő technológia- és eszközválasztás, valamint ezek használati módja együttesen határozzák meg.

2.3. Célkitűzés

Az informatikai projektek alapvető tervezési és felelősségvállalási hiányosságaira való rámutatást, valamint a nevezett kérdéskörben már több évtizedes hagyományokkal rendelkező műszaki szakágak gyakorlatának bevezethetőségi vizsgálatát tűzte ki célul a jelen tanulmány. Más műszaki szakágak alatt főleg a Hírközlési, valamint Elektrotechnikai és Épületvillamossági területekre kell gondolni, mint a jó mérnöki gyakorlat mintaadóiként, és mint az informatikához legközelebb álló területekként. Elsődlegesen a problémákat orvosolni lehetne a jogosultságrendszer bevezetésével a tervezési oldalon, valamint az ehhez kapcsolódó dokumentációk (koncepcionális, kiviteli, megvalósulási dokumentációk), valamint a tervezői nyilatkozat kötelező elemként történő definiálásával. Ez utóbbi a felelősségvállalás és szabványi megfelelés szempontjából kiemelten fontos elem, amellyel általános minőségi színvonal emelkedést lehetne elérni, természetesen a felelősségvállaló tervező által aláírt és jóváhagyott tervekkel egyetemben. A dokumentációk előre történő elkészítése, a projekt műszaki tartalmának az implementálás előtti tervezése elengedhetetlen bármely más műszaki terület nagy

projektjeihez hasonlóan. Továbbá az ok-okozati összefüggéseket az informatikai projektek esetében is következetes módon kell a gyakorlatba ültetni, ami esetünkben a tervezés, jóváhagyás, kivitelezés hármásával történhet meg. Célként tehát meg kell fogalmaznunk, hogy a tervezési és kivitelezési gyakorlatok/folyamatok beemelése, adaptálása kell, hogy elsődlegesen megvalósuljon, a korszerű módszertanok alapján. Ez természetesen nem jelent egy az egybeli megfeleltetést, hiszen az informatikai projekteknek mások a dokumentumai, kritériumai és sok esetben a munkamódszerei is, mint adott esetben egy építőipari beruházás kivitelezése közben, egy építész, vagy villamossági területen. Ugyanakkor az analógiák megtalálhatók, és a mögöttes tartalom, illetve a cél ugyanúgy beazonosítható, úgymint a felelősségvállalás, és az áttekinthető, letisztult dokumentáció.

3. Szakmai minőségi célok az informatikai projekteken

A közepes és nagyobb méretű IKT projektekkel szembeni alkalmazási – felhasználói - szempontú (továbbiakban üzleti) követelmények szerteágazó volta, a mind jobban előtérbe kerülő élet és vagyonvédelmi szempontok, ebből eredően a rendszerek bonyolultsága miatt prognosztizálható a megvalósult megoldás szállítótól független szervezet(ek) általi minőségellenőrzésének, auditjának, később az engedélyezési eljárásának a szükségszerűsége.

Várható, hogy a technikai-technológiai fejlődéssel párhuzamosan, a biztonsági igények és az életvédelmi kockázatok növekedésével ki fog majd alakulni az ilyen típusú ellenőrzések intézményesített (hatósági) változata, nem csak az új „létesítmény” üzembe helyezésekor, hanem akár időszakonként is meg kell majd újítani a korábban megszerzett üzemeltetési engedélyeket.

A szakmai minőségbiztosítás és a minőségellenőrzés fő tevékenységei:

- 1.) Hardver és konfiguráció megfelelőség vizsgálat.
- 2.) Alkalmazás funkcionális megfelelőség ellenőrzés.
- 3.) Biztonsági és megbízhatósági megfelelőség vizsgálat.
- 4.) Terheléses és teljesítménytesztelés.

A fenti tevékenységek önállóan is elvégezhetőek, ugyanakkor az egyes megállapítások kihathatnak a többi vizsgálati szakterületekre is. Ezeknek egy-egy típusát, változatát, a partnerek közötti vállalkozási szerződésnek megfelelően, ma is elvégzik a fejlesztő-megrendelő relációban önállóan vagy az átadásoknál-átvételeknél mindkét fél jelenlétében.

Ugyanakkor kitapintható az igény a minőség-ellenőrzés szállítótól független szervezet általi elvégzésére. Ma még nincs ennek jogszabályi kötelezettsége, a független szervezet általi szakmai minőségellenőrzés még nem széleskörűen alkalmazott, bár erre a feladatra is egyre több nagyprojekt alkalmaz külső szakértőket vagy szakértő cégeket: az EU alapokból finanszírozott projekteknél ez már elvárás, és például az operatív minőségellenőrzési feladatok ellátására a kormány ki is jelölte a KIFÜ szervezetét.

Az IoT (Internet of Things) rohamos terjedésével mind több eszközünk, gépünk irányítását fogjuk informatikával támogatva megoldani, ezért szükséges idejében felkészülnünk erre az időszakra. Várhatóan az engedélyezési eljárások a gyakorlatban fokozatosan fognak – a tapasztalatok visszacsatolásaként – kialakulni.

Az audit jellegű ellenőrzési módszereknek, eljárásoknak publikusnak kell lenniük, így az érdekeltek is előre elvégezhetik majd azokat.

Megállapíthatjuk, hogy a fentnevezett négy vizsgálati terület közül ma az alkalmazás funkcionális megfelelőség vizsgálata és ellenőrzése a legesetlegesebben végrehajtott terület, itt a legkevesebb a mindenki által elfogadott, illetve kötelező szabály, módszer stb., Ma a gyakorlatban erre a célra a funkcionális követelményspecifikáció alapján kialakított tesztelési tervek elkészítése adhat némi garanciát.

Annak érdekében, hogy az alkalmazások funkcionális megfelelőség ellenőrzési területén is előre tudjunk lépni, állítottuk össze a jelen anyagunkat ajánlási céllal, vitaindító jelleggel.

3.1. Szemlélet, megközelítés

Véleményünk szerint az informatikai rendszerek minőségének javításában a szemléletváltás, egy új megközelítés hozhat jelentős változást.

A megfelelés minőségének ellenőrzéséhez fontos (újra)értelmezni a főbb jellemzőiket, hogy ezek mentén egységes, mérhető, pontos minőségi elvárások legyenek megfogalmazhatók velük szemben. Ezért itt sorra vesszük az informatikai rendszerek fő jellemzőit, kifejtjük helyüket és szerepüket a létrehozásuk folyamatában, majd a továbbiakban, e rendszerek fő fejlesztési lépései mentén, a minőség ellenőrzéséhez megadunk néhány általunk fontosnak tartott szempontot, eljárást, és technikát.

Szemléletünk szerint az informatikai rendszerek információ technológiai eljárásokkal létrehozott termékek, emberi munkafolyamat automatizáló "célberendezések", amelyek fizikai eszközök (számítógépek), és rugalmasan alakítható emberi szabályok (programok) segítségével adatokból, feldolgozási folyamatokon keresztül, "információt" állítanak elő. Az ilyen rendszerek tehát szabály alapon működő „gépek”, amelyek bemenete adat, és kimenete, az emberi szubjektum segítségével kialakított, sokrétű információ.

Ebben a meghatározásban adatnak az önmagában nem egyértelműen értelmezhető ténytet tekintjük, amiből információ az emberi értelmezés eredményeként áll elő, és ez egy szoros kapcsolatot épít ki az adatkezelő rendszerek, és az azt jelentéssel felruházó, információvá alakító felhasználóik között. Ezzel az „értelmező ember” - gép kapcsolattal együtt nevezhető csak egy rendszer "informatikainak".

Az adatfeldolgozási folyamat, ami az adatok befogadását, létrehozását, átalakítását, és bemutatását jelenti, az ember alkotta szabályok (logikák) összessége.

Végző soron magát az informatikai rendszert egy különleges eszköznek („szerszámnak”, „gépnek”) tekinthetjük, amelyet emberi fogalmak (adatok) kezelésére („megmunkálására”) használhatunk. Mivel a fizikai (anyagi) korlátok az ilyen rendszer által kezelt emberi fogalmakra nem értelmezhetőek, a fogalmak pedig a fizikai hordozó közegtől (a számítógép) tartalmukat tekintve függetlenek, ezért az informatikai rendszer ma az egyetlen olyan „szerszám” az emberiség kezében, amelynek alakíthatósága, felhasználhatósága szinte „végtelen”.

Az emberi tevékenységet segítő különféle eszközökkel összehasonlítva, az informatikai rendszerek specifikus jellemzője, hogy legmagasabb szintű elemeik, lényegi alkotórészeik (adatok, adatkezelő szabályok) az emberi gondolkodás, és értelem termékei, azaz fogalmak. Hasonlóan az emberi agy működéséhez, ahol a biológiai (természeti) alapú szervezet, és azon belül elsősorban az idegsejtek pusztán fizikai működésével nem írható le a rájuk épülő tudat, ami így egy új minőség, ugyanígy az informatikai rendszerek is, habár természettudományos törvényeken alapuló fizikai eszközökön, számítógépeken működnek, mégis új, csak az emberi elmével megfogható szintet képviselnek. Az ilyen rendszereket működtető programok emberi szabályok, amelyek összehangolt működését az ember alkotja meg, a számítógépek pedig ezen szabályok precíz, megbízható, gyors végrehajtását segítik. Fontos jellemző, hogy a természeti világ objektív, fizikai törvényeivel szemben itt az emberi értelem, felfogás, probléma megoldás, tudati működés dominál, ezért az informatikai rendszerek kialakításának sarkalatos pontja az értelmezés, a fogalmi gondolkodás és a logika. Új informatikai rendszer létrehozásakor különösen szembevetendő a fenti állítás megalapozottsága, hiszen a vele szemben támasztott felhasználási igények megfogalmazása egy adott gondolatkörben, kontextus térben jönnek létre, amit azután meg kell feleltetni az információ technológiai fogalom-tér elemeinek ahhoz, hogy a megoldás, az informatikai rendszer létrejöjjön. Az informatikai rendszerekre tekinthetünk úgy, mint fogalmi modelleken alapuló tudás bázisokra.

3.1.1. Információ technológiai sajátosságok

Alapvető jelentőségű a rendszerek minőségének megítélésében a magas szintű technológiai jellemzők ismerete. Az alábbiak segítenek megalapozni a megközelítésüket.

3.1.1.1. Működési logika, mint eredmény termék

Az informatikai rendszerek létrehozásának folyamata, technológiájuk belső lényege - az adatok információvá alakítása - egy értelmező, elemző-szintetizáló modell kialakítása, amiben az ember tudati szemlélete dominál. A szemléletmódok

egyéneenként eltérők lehetnek, így összehangolásuk, az információ "megjelenítése", felfogásának megkönnyítése sokféle megoldási lehetőséget kínál. Ráadásul egy-egy szemléletmód kialakítása, a fizikai szinten megvalósuló példátlan függetlenség (a programkódok, és adatok fizikai szinten csak egyszerű, az adat-, és a program szempontjából azonos jellemzőkkel bíró elektronikus jelek) miatt, összehasonlíthatatlanul rugalmasabb, összevetve az anyagi természetű elemeket felhasználó tevékenységekkel. Az informatikai rendszereknek azonos iparágon belül is jelentősen eltérő megoldásai lehetnek, és ennek éppen ez a lényegi sajátosság az oka, egyben emiatt nincs átütő eredménye sem, a más területeken megszokott, és jól működő szabványosításnak.

3.1.1.2. Tervezés

Az információs rendszerek összetettsége a gyakorlat során megkívánja ezt a bonyolultságot hatékonyan kezelő meg-, vagy felosztás alkalmazását, hasonlóan más, szintén összetett tevékenységek jól megszokott kezeléséhez.

Az iparági megoldások logikailag, és erre épülve fizikailag is rétegzettek (layer, tier). E rétegzettség előnye, hogy - kis túlzással élve - bármelyikük cserélhető, ami a technológiai fejlődést igen jól támogató jellemzőjük.

A korszerű, de facto megoldások manapság legalább három, viszonylag önálló, de egymással kapcsolatban álló réteget különböztetnek meg:

- Adat megőrző réteg, ami az adatok, és a közöttük értelmezhető belső összefüggések tárolását végző (persistence) modul.
- Adatfeldolgozás vezérlő réteg, ami az "üzleti"- (kezelési), hozzáférési-, továbbítási-, tehát az adatokat alakító, elrendezésüket megvalósító (middle-, vagy business tier) modul.
- Adat megjelenítő réteg, ami az eredményt megosztó, a felhasználónak szóló megjelenítést végző (presentation) modul.

3.1.1.3. Munkamódszerek

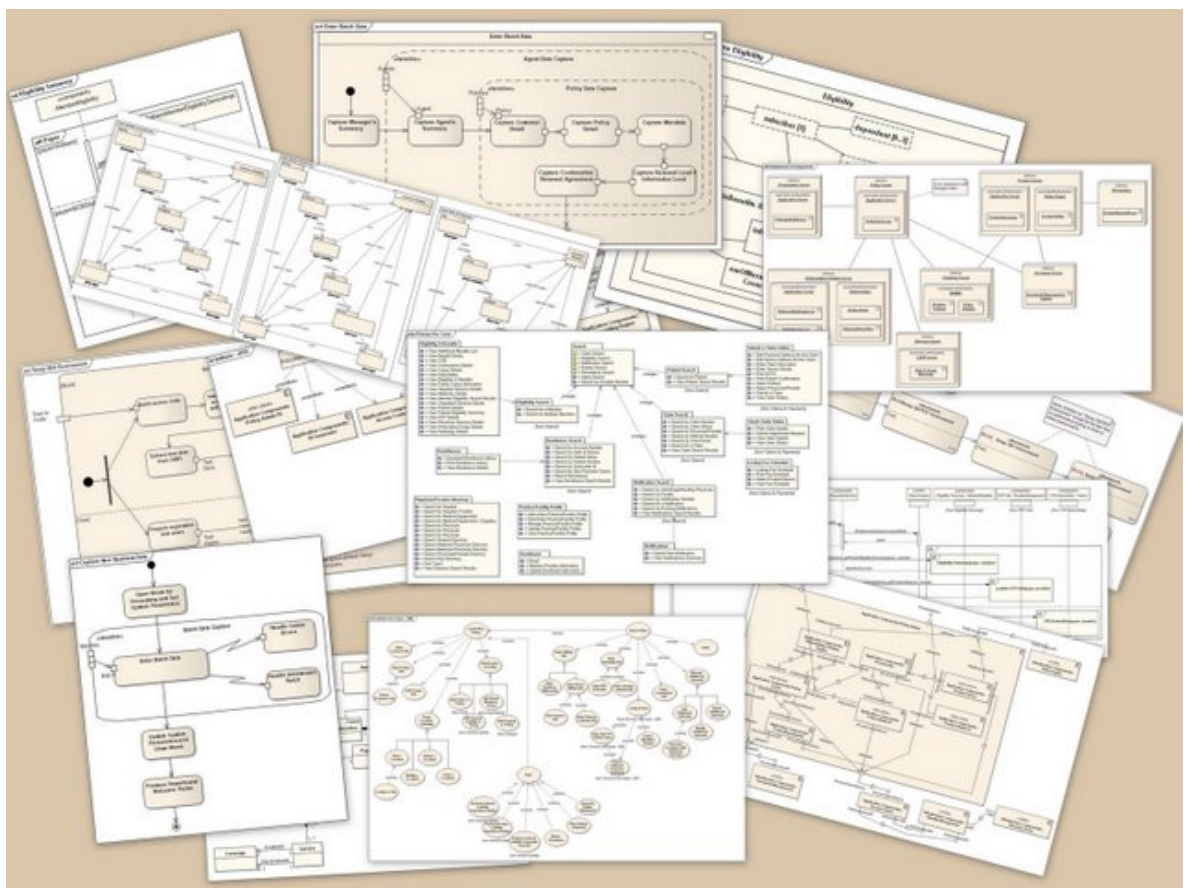
Az alábbiakban, a napi gyakorlat során követendőnek tartott megállapításokat teszünk a rendszerek építési elveivel, és az építési tevékenység jellemzőivel, szabályaival kapcsolatban.

3.1.1.3.1. Rendszerépítés

Az informatikai rendszerek építése tervezés, modellezés útján megvalósított tevékenység, és mint más műszaki területeknek, ennek is vannak terv típusai. Az

ilyen rendszereknek fizikai, és logikai felépítése is van. Az előbbit, a rendszert megvalósító eszközpark alapján, a mérnöki tervekhez hasonlóan, a fizikai elhelyezkedés és berendezés jegyzék alkotja. A logikai elrendezés, és a működés leírása már összetettebb, és az iparágra jellemzően specifikus, ún. nézet, vagy szempont alapú.

Az adat-, és feldolgozás elemek leírását statikus nézetek alkotják, melyek szerkezeti, és kapcsolati modellek. A feldolgozások belső-, és folyamatokba rendezett működését dinamikus modellek sora írja le, amelyek a működési jellemzőket eltérő nézőpontokból mutatják be. A modellek használatára nincs szabvány, vagy szabály, mindig az érthetőség szabja meg a használt modellek típusát, és részletezettségét, míg a modellek ábrázolási jelkészletének jelenleg az UML modellező nyelv az elfogadott, de facto eszköze. Az egyes szintek, és témakörök specifikációját különböző dokumentumtípusok rögzítik, ezeknek prototípusait a munkánk során elkészítettük, és hozzáférhetővé tettük.



1. ábra: UML diagramm paletta

3.1.1.3.2. Munkaszervezés

A korábbi bekezdésekben megfogalmazott sajátosságok miatt általánosan jelentkező kivitelezési problémák feloldására számos áthidaló, főként munkaszervezési újításokon alapuló módszer jött létre. Lényegüket tekintve az általunk javasolt első projekt fázisban megvalósítandó elemzést “csempézik be” a teljes, hagyományos folyamatba, vagy nyújtják el, annak majdnem teljes hosszában.

A fenti megállapításoktól egyetlen eltérő, a legkorábbi, és mára már biztosan kijelenthetően csak nagyon speciális esetekben használt “vízesésszerű” munkamódszer lényege, hogy a fejlesztési életciklus lépéseinek sorrendjében halad végig az eredménytermékek teljes vertikumában. Minden lépés végén lezárja azt, és csak ezután kezdi el a következőt. Így az eredmények egyszerre készülnek el, a teljes folyamat végén. A mai elvárásoknak ez a módszer szinte semmilyen jellemzőjével nem felel meg, ennek ellenére még mindig ezzel a szemlélettel, vagy ennek bázisán készülnek a projektek hazánkban, ami súlyos elmaradottságnak számít a nemzetközi trendekkel, legjobb gyakorlatokkal összevetve.

Az “iterációs” módszerek lényege, hogy kiragadnak egy eredmény terméket, vagy termék részt a megvalósítandók közül, és azon végig futtatják a fejlesztési életciklus lépéseit, az elemzéstől, a tervezésen át, a kivitelezésig, és ezt teszik minden további elemmel, elem résszel, egészen addig, amíg sorra nem kerül az összes eredmény termék.

A ma divatos "Agilis", vagy "Scrum" tervezési, kivitelezési, illetve fejlesztés irányítási módszerek is ilyenek, azonban csak kisebb ráfordítású, jól körülhatárolt célok megvalósításánál alkalmazhatók sikerrel, az elképzelés alapját képező fokozatos rendszerbővítés, mint munkamódszer miatt. A többszörösen összetett rendszereknél, az alapvető rendszerelemek, és funkciók korai meghatározása, és ezzel a technológiák kiválasztása adja a többi funkcionalitás alapját, így ezeknek egy később felismert helyzet miatti jelentős átalakítása a rendszer teljes újratervezéséhez vezethet.

3.1.1.4. Ráfordítás meghatározás

Sajátságosan, minden informatikai rendszer önmaga prototípusa, és egyben maga a “késztermék” is. Ez eleve megnehezíti a létrehozásukhoz szükséges, akár csak közelítő ráfordításbecslést is. A prototipizálás lényege ugyanis a folyamatos alakítás, kísérletezgetés, finomítás, a késztermék viszont a prototípus tapasztalataiból létrejövő, megbízható, működő példány. E két ellentétes munkamódszer együttes hatása nagy terhet ró, a ma általánosan elfogadott, projektszerű kivitelezési megközelítésre, amely alaptermészete szerint eszközök, és jól ismert kivitelezési

feladatok összeválogatása alapján, az azokhoz jól meghatározott munkavégzési idők (egységnyi teljesítmények) összevetésével előre próbálja meghatározni egy eredménytermék eléréshez szükséges ráfordításokat, vagyis a fizikai (építési) világból veszi a hasonlóságot. Ugyanakkor, ahogy azt már korábban megállapítottuk, az informatikai rendszerek létrehozásának középpontjában jelentős mennyiségű, és összefüggésű működési szabályok megállapítása, és együttműködésük meghatározása, létrehozása áll, vagyis egy fogalmi-eljárási modell létrejötte a cél, aminek fókuszában az információ előállítás van. Az pedig, hogy mi tekinthető információnak, szemléletbeli kérdés, és mint ilyen, előzetesen, a feladatot generáló igényekben inkább csak felületesen, körülíró módon jelenik meg, érdemben viszont csak később, a kialakítása közben határozható meg. Természetes következménye az ezt a jellemzőt elhanyagoló munkamódszernek, hogy a ráfordítások tervezett, és tényleges értékei jelentősen eltérnek egymástól. Általános probléma, hogy az erre megoldást kínáló, a végtermékek meghatározásából kiinduló módszer sem vezet eredményre, mivel az "információvá válás" egy folyamat eredménye, ami maga az információs rendszert építő projekt lényege, így lehetetlen előre pontosan meghatározni a végtermékek tartalmát, és főképpen a mennyiségét. Gyakori, hogy csak az elemzési-tervezési folyamat részeként határozható meg pontosan az alkalmazandó géppark, és a felhasználni kívánt technológiák.

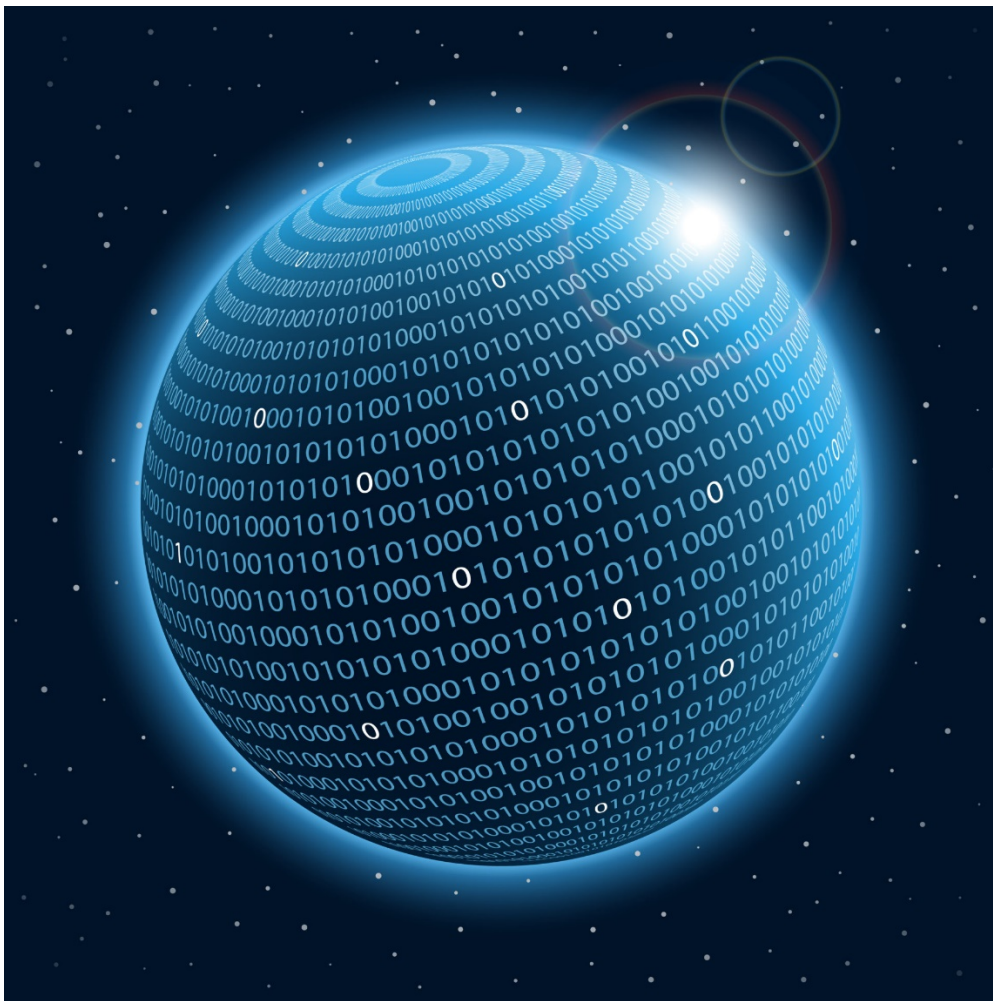
Nézetünk szerint a megoldás a projektek kétfázisú megvalósítása lehetne. Az első fázis egy előkészítő projekt, aminek a célja az eredménytermékek minél pontosabb meghatározása, egy, a projekt teljes időtartamához viszonyítva jelentős hosszúságú elemzési periódusban, amely a hangsúlyt az adatok értelmezésére, az információ tartalmi elemeinek meghatározására helyezi, és egy pilot jellegű, az elvárt fő funkcionalitást plasztikusan tükröző megoldás létrehozását is magában foglalja. A második, tervezési-megvalósítási fázis előtt így lehetővé válna a pontos ráfordítás meghatározása. Természetesen a jelenleg általánosan elfogadott finanszírozási szempontú megközelítés nem konform ezzel az elgondolással, de emlékeztetünk rá, hogy az informatikai, vagy bármely más típusú fejlesztés elsősorban nem pénzügyi kérdés. A gazdasági szakterületek ma megfigyelhető, hegemónikus uralmát a mindennapi élet döntéshozatalaiban mindenképpen átgondolandónak tartjuk. Nézetünk szerint világunkat nem a gazdasági, hanem az eszmei megfontolások viszik előre, a gazdasági területek szaktudása ugyanolyan támogató jellegű ebben a fejlődési folyamatban, mint maga az informatika is, így az ilyen megfontolások helye, szerepe, és súlya sem lehet túlzott, vagy az emberi fejlődést akadályozó jellegű.

3.1.1.5. Egyéb jellemzők

A fizikai világ anyagi korlátaitól való (relatív, de mindenképpen szabadság fokban abszolút értelmű) mentessége miatt, az informatikai rendszerek az előállításukhoz viszonyítva rendkívül alacsony befektetéssel többszörözhetők.

A közvetítő közeg (elektromosság) sajátosságai miatt térbeli mozgásuk, áthelyezhetőségük a leggyorsabb a többi eszközhöz viszonyítva.

Változtathatóságuk, összekapcsolhatóságuk lehetőségei elméletileg korlátlanok a közös platform, a digitális „alapanyag” miatt, de ráfordítási szükségletük - ma még - ezzel nem arányos.



1. ábra: A Digitális Világ, az ember által teremtett Virtuális Valóság

3.1.2. Technológiai modellek

A jelenkori számítástechnikai fejlettség mind fizikailag, mind logikailag a nagyfokú elosztottságot, rétegzettséget támogatja, sőt, akár mindezek elfedését is lehetővé teszi, a virtuális térbe helyezett, és így a fizikai lokalitást, valamint a logikai megosztottságot akár egyszerre is elrejtő, szolgáltatás alapon működtethető technológiák rohamos terjedésével.

A korábban említett, természetszerűen sokféle felépítésű, működésű informatikai rendszerek mellett léteznek, ma már hagyományosnak nevezhető működési, vagy architekturális modellek is, amelyek mintaként szolgálhatnak informatikai megoldás fajták azonosításakor, vagy éppen adott típusok fejlesztésekor.

Ezek, vázlatosan a következők:

- **Tranzakciós modellek**
Az operatív munkavégzés információit kezelő rendszerek.
- **Adattárház modellek**
Információs rendszerek adatait integráló, elemzést, és döntést támogató rendszerek.
- **Web technológia alapú modellek**
Intranet vagy Internet hálózaton keresztül elérhető, speciálisan elosztott felépítésű rendszerek.
- **“Felhő” technológia alapú modellek**
Csak az Interneten keresztül elérhető, a felépítésüket elrejtő, és szolgáltatás jelleggel működő, földrajzilag globálisan elérhető rendszerek.

Ezek mellett létezik részletesebb tipizálás, besorolás is, amely az operatív feladatvégzést támogatótól, az elemzői, szakértői, stratégiai döntéshozói, illetve a tudás alapú, legújabbban pedig a mesterséges intelligencia rendszerekig terjed.

3.1.3. Szabványosítás és mindennapi gyakorlat

Visszautalva a korábbi megállapításokra, a szabványosítás az informatikai rendszerek belső természeténél fogva szórványos, és ebben nem is várható gyors változás. Meglátásunk szerint a szabványok hiányának oka az értelmező, elemző-szintetizáló jellegű munka eredményeként előálló “szemlélet”, ami egyébként az adott informatikai rendszerben ölt testet, és ami alkotójának gondolkodását tükrözi, vagyis

azt, ahogyan megértette, és elképzelte az adatkezelési, információ létrehozási folyamatokat. Ilyen módon nincs két egyforma rendszer, nincs két egyforma megközelítés vagy megvalósítási architektúra. A szabványosítás itt a legjobb gyakorlatok, a tervezési, megvalósítási technikák, módszertanok, és minták megosztását, illetve iparági minta architektúrák („Telco model”, „Banking model” stb.) desztillálását jelenti csupán. A legnagyobb ilyen szabvány-bázissal az adatmodellezés rendelkezik, a maga egzakttsága miatt (ANSI, ISO szabványokkal támogatottan).

Egyes, jól körülhatárolható technikák, technológiák azonban kifejezetten szabvány alapúak, különben nem lehetne együttműködés a különböző rendszerek között.

3.1.3.1. Az informatikai rendszerek reprezentációja

Tapasztalataink szerint bizonytalanság, és ebből következően nagyfokú kuszaság fedezhető fel a mai informatikai rendszerek leírásában, bemutatásában. Nézőpontok, hangsúlyok keverednek egymással, sokszor hiányzik a megfelelő részletezettségű specifikáció, a formátumok, és az elnevezések sem közkeletűek, vagy éppen azok, de tartalmukban nem adekvátak, és sem a készítő, sem a megrendelő nem elégedett az eredményekkel, ami jelentősen visszaüt a rendszerek (mind létrehozási, mind üzemeltetési) ráfordításaiban.

Véleményünk szerint már a tervező munkát is jelentősen segítené egy olyan szempontrendszer és kategorizáció, aminek alkalmazásával természetes módon kerülne fókuszba minden, egy konkrét rendszerre alkalmazható releváns aspektus, ami teljessé, és jó minőségűvé tehetné mind az eredmény termékeket, mind azok specifikációit.

A rendszerek leírásához segítségül használhatjuk a 35 éves múltra visszatekintő Zachman besorolási rendszert, egy átfogó, ontológiai modellt, ami akár egy tárgy, akár egy elképzelés teljességre törekvő (holisztikus) körülírásának, bemutatásának módját rögzíti.

A besorolási rendszert az informatikára alkalmazva, az alábbi specifikációk, és nézőpontok állnak elő:

	DATA <i>What</i>	FUNCTION <i>How</i>	NETWORK <i>Where</i>	PEOPLE <i>Who</i>	TIME <i>When</i>	MOTIVATION <i>Why</i>
Objective/Scope (contextual) <i>Role: Planner</i>	List of things important in the business	List of Business Processes	List of Business Locations	List of important Organizations	List of Events	List of Business Goal & Strategies
Enterprise Model (conceptual) <i>Role: Owner</i>	Conceptual Data/ Object Model	Business Process Model	Business Logistics System	Work Flow Model	Master Schedule	Business Plan
System Model (logical) <i>Role: Designer</i>	Logical Data Model	System Architecture Model	Distributed Systems Architecture	Human Interface Architecture	Processing Structure	Business Rule Model
Technology Model (physical) <i>Role: Builder</i>	Physical Data/Class Model	Technology Design Model	Technology Architecture	Presentation Architecture	Control Structure	Rule Design
Detailed Representation (out of context) <i>Role: Programmer</i>	Data Definition	Program	Network Architecture	Security Architecture	Timing Definition	Rule Speculation
Functioning Enterprise <i>Role: User</i>	Usable Data	Working Function	Usable Network	Functioning Organization	Implemented Schedule	Working Strategy

1. ábra: Zachman kategorizációs rendszere, informatikai fejlesztésre alkalmazva

A modell elméleti lényege egyszerre hétköznapi, és filozofikus. Bármely dolog, akár fogalom, akár valós fizikai tárgy, jelenség meghatározása objektív, és szubjektív jellemzőkkel írható le.

Az objektív leíró jellemzők a "Mit?, Hogyan?, Hol?, Ki?, Mikor?, Miért" kérdésekre adott válaszok összessége, a szubjektívek pedig azoknak a nézőpontoknak megfelelő válaszok, amelyekkel az adott dologra tekintünk, egyfajta hangsúly, lényegkiemelés szerint. Például az informatikai rendszerekre másként tekint, más részletezettséget, információt jelent a felsővezetőnek, a rendszer tervezőjének, a programozónak, az azt üzemeltetőnek, és a végfelhasználónak. Röviden tehát, az összetett dolgokról, a rájuk tekintő szubjektum, a megközelítés célja alapján, tartalmában és típusában is eltérő (szöveg, grafika) leírást adhat. A teória 36 osztályozási lehetőséget kínál, amivel - állítása szerint - bármi, a maga teljességében leírható, bemutatható. A bemutatási kategóriák 6 nézőpont szerint leírható absztrakciós tartalmat jelentenek, és mindegyik tartalom átvihető a másikba, átalakító szabályok alkalmazásával úgy, hogy nem a részletezettség változik nézetenként, hanem csak megfeleltetés történik a különböző absztrakciós szintek között.

A mátrix minden sora egy adott szempontnak megfelelő "teljesség", egy absztrakciós szintű leírása az elérni kívánt dolognak, ami pontosan a szempontnak megfelelő részletezettségű, mélységű stb. Nincs minőségi különbség a nézetek (sorok) között, mindegyik a maga nemében teljes, hiánytalan. A teljesség azt jelenti, hogy az "eredmény" leírása pontosan a szükséges, és egyben elégséges módon történik. Az

osztályozási rendszer arra is választ ad, hogy egyik elemének változása hogyan hat az egész rendszerre. A modell teljes kifejtésére a jelen írás keretében nincs lehetőségünk, de ez elérhető, tananyag szintjén is, a felsőoktatásban, és számos leírás található róla az interneten is.

A nagyvállalati információs rendszerekre (Enterprise Architecture), mint absztrakt jelenségre alkalmazva ezt a modellt, a különböző témakörökhöz létrehozott dokumentum típusainkat a besorolási módszer alapján helyeztük el:

Zachman-féle kategorizálás	Mit ?	Hogyan ?	Hol ?	Ki ?	Mikor ?	Miért ?
Kiterjedés (Környezeti) MEGBÍZÓ	Követelmény specifikáció	Üzleti folyamat specifikáció	Lokáció specifikáció	Érintett szervezetek	Esemény specifikáció	Üzleti célok
Üzlet modell (Áttekintő) TULAJDONOS	Tartalmi modell	Üzleti folyamat modell	Munkaszervezési rendszer	Munkafolyamat modell	Időzítési modell	Üzleti terv
Rendszer tervezés (Logikai) TERVEZŐ	Logikai modell	Logikai Rendszerterv	Rendszer architektúra elrendezési modell	Humán Interfész specifikáció	Feldolgozás felépítés	Üzleti szabály modell
Kivitel tervezés (Fizikai) KIVITELEZŐ	Fizikai modell		Technológiai architektúra modell	Megjelenítési architektúra	Vezérlési elemek	Szabály terv
Gyártás (Megvalósítási) KÉSZÍTŐ	Adat-, és folyamat definíció	Szoftver program	Hálózati, és funkcionális egységek kiépítése	Biztonsági architektúra	Időzítési definíciók	Szabály specifikáció

4 ábra: Az ajánlott specifikációs sablonok informatikai besorolási rendszere

3.2. Az információs rendszerek tervezésének és fejlesztésének módszertana

Ebben a fejezetben áttekintjük az információs rendszerek tervezésének és fejlesztésének ma ismert, széles körben alkalmazott, és kvázi szabványosnak tekinthető módszereit, főbb lépéseit és dokumentumait abból a célból, hogy megvizsgáljuk, hogy melyek azok az alapfeltételek, amelyek a minőségi célok eléréséhez feltétlenül szükségesek.

Az információs rendszerek belső sajátossága, az adatok felhasználási szempontú szabályokon alapuló elrendezésének és átalakításának alapvetően szubjektív természete miatt, a felhasználói elégedettség, mint elsődleges mérce mellett, a minőség biztosítása legalább az alábbi módokon ellenőrzendő, és támogatandó:

- Projekt, vagy operatív minőség-ellenőrzés: elsősorban az adminisztratív/időbeli, elszámolásbeli/dokumentációs megfelelőséget vizsgálja. Ez EU-s projekteknél már jogszabályi előírás, amit a KIFÜ (Kormányzati Informatikai Fejlesztési Ügynökség) végez.
- Szakmai minőségbiztosítás: a követelményeknek történő megfelelőséget vizsgálja, beleértve a funkcionális megfelelőséget is. A projekt lezárásával ez a tevékenység lezárul.
- Engedélyezési audit: ez egy ma még hiányzó, az engedélyezéshez és fenntartáshoz szükséges audit, az utólagos minőség-ellenőrzés.

Az eddig felsorolt, speciálisan az informatikára jellemző tulajdonságok, és adottságok alapján, valamint a megvalósítandó rendszerre vonatkozó aktuális elvárások mentén egy konkrét terv formálisan összevethető a megvalósításához használt módszerekkel, eljárásokkal, eszközökkel, és jó hatásfokkal megállapíthatók a tervezési/kivitelezési problémák.

Alapvető cél az anomáliák korai detektálása, ezért ezt az összevetést az informatikai projektek teljes életciklusában, folyamatosan kell végezni.

A következő fejezetek, a rendelkezésre álló terjedelem korlátozottsága miatt, néhány magas szintű viszonyítási pontot mutatnak be, amelyek mérceként alkalmazhatók egy-egy terv megítélésénél.

3.2.1. Információs rendszerek tervezési fázisai

Az informatikai rendszerek tervezése alapvetően perspektivikus, vagyis számos nézőpont alapján jön létre a teljes megoldási modell, az összkép. Elengedhetetlennek tartjuk annak hangsúlyozását, hogy több, eltérő szempontú modellel lehet csak

biztonságosan leírni egy ilyen rendszert, és a hangsúly nem csak az adatok modellezésén van, hanem nagyobb nyomatékkal a folyamatokon, a működésen, működtetésen. Ez utóbbi jellemzők az (új) szemlélettel foglalkozó fejezetben bemutatott különleges lényegi tulajdonságok megragadása miatt fontosak. A szabályok megvalósítása a programkódokban jön létre, és ezek végrehajtási sorrendje, függése, kapcsolata, együttműködése a működő rendszerben igen összetett folyamat környezetet teremt. Ebben a "világban" a folyamatok lezajlása, inputjaik, és outputjaik megjelenése, a hibák kezelésének módjai csak jelentős mennyiségű, és jól megszerkesztett nézőpontú ábrázolással darabolhatók fel, az ember számára kezelhető részekre. Az ilyen dinamikus modellek működést szimuláló bemutatására is képesek már a tervező eszközök, így mintegy "kipróbálható", ellenőrizhető a tervezett működés helyessége.

Az Objektum Orientált alapon szervezett rendszereknél egyenesen elképzelhetetlen pusztán szöveges útmutató alapján, a percenként keletkező, megszűnő, és többnyire egyszerre létező objektumok bonyolult kommunikációs kapcsolatának, adott funkciót megvalósító hálózatának felfogása, kezelése.

További indoka lehet a szemléletes, modellel történő ábrázolási megközelítésnek az elosztott rendszerek által létrehozott összetett, rétegzett megoldásokban, az egyes szabványok egymásba alakulásának, kommunikációjának tervezése, nyomon követése, tesztelése.

3.2.1.1. Követelmény kezelés

A Követelmények a rendszerrel szemben támasztott, célzottan, pontosan definiált elvárások, amelyek teljesülése alapvető mutatója az eredménytermék „jóságának”, elfogadhatóságának, és egyben biztos alapja a tesztelésének is.

A követelmények osztályozása (vonatkozási) tárgyuk alapján az alábbi:

1. Üzleti követelmények

Az informatikai rendszerrel megoldani kívánt probléma jellemzői, elvárások, és célok formájában.

2. Funkcionális követelmények

Az alkalmazási rendszer (szoftver) elvárt „tudásának”, mint funkciók összességének a tételes felsorolása. Fő jellemzőjük a „mérhetőség”, azaz a konkrét, pontos nevesítés. Kategorizálásuk elengedhetetlen, mert ez kifejezi

megértésüket, és kezelésük, felhasználásuk pontjainak helyes beazonosítását. Az alábbiak a főbb Funkcionális követelmény kategóriák:

- **Szervezeti követelmények:** Szervezeti házirendek, belső szabályzatok.
- **Törvénykezési követelmények:** a rendszer által teljesítendő törvények, és hivatalos elvárások, szabályozók.
- **Etikai követelmények:** a biztonsággal, személyes adatok védelmével kapcsolatos rendelkezések.
- **Leszállítási követelmények:** a rendszer leszállításának, és terjesztésének szabályai.
- **Szabvány érvényesítési követelmények:** a rendszer fejlesztését és működését/működtetését leíró előírások.
- **Külső követelmények:** külső szereplővel kapcsolatos szabályok, például integráló rendszer befogadási szabályai.

3. Nem funkcionális követelmények

A rendszer hatékonyságával kapcsolatos, többnyire nem mérhető jellemzőkkel definiált elvárások. Például a rendszertől elvárt minőséget leíró követelmények, igények.

4. Szabályok

Technikai, vagy egyéb, pontosan megfogalmazott, kötelezően betartandó korlátok, amelyek a rendszer fejlesztésének, illetve magának a rendszernek bármely részletére vonatkoznak.

3.2.1.1.1. Követelmény specifikáció

A követelmények meghatározása egy „közelítő” eljárás sorozat, amelynek eredménye egy tételes jegyzék, osztályozott felsorolás. A leírásoknak pontosan, a félreértéseket, félreértelmezéseket a lehető legjobban elkerülő módon kell kifejtetniük az egyes követelmény tételekben hivatkozott fogalmakat. A specifikáció mélységét elsődlegesen a téma fontossága, precizitásának szükségessége, illetve összetettségének mértéke határozza meg, így az egyes tételek részletezettségi szintje (kifejtése) eltérő lehet.

A Követelmények részletezettségi igényei, illetve a vonatkozó témakörök jellemzői alapján több modellezési technika áll rendelkezésre:

- Igény specifikáció
- Esettanulmány modell
- Állapotgép modell

3.2.1.1.2. Igény specifikáció

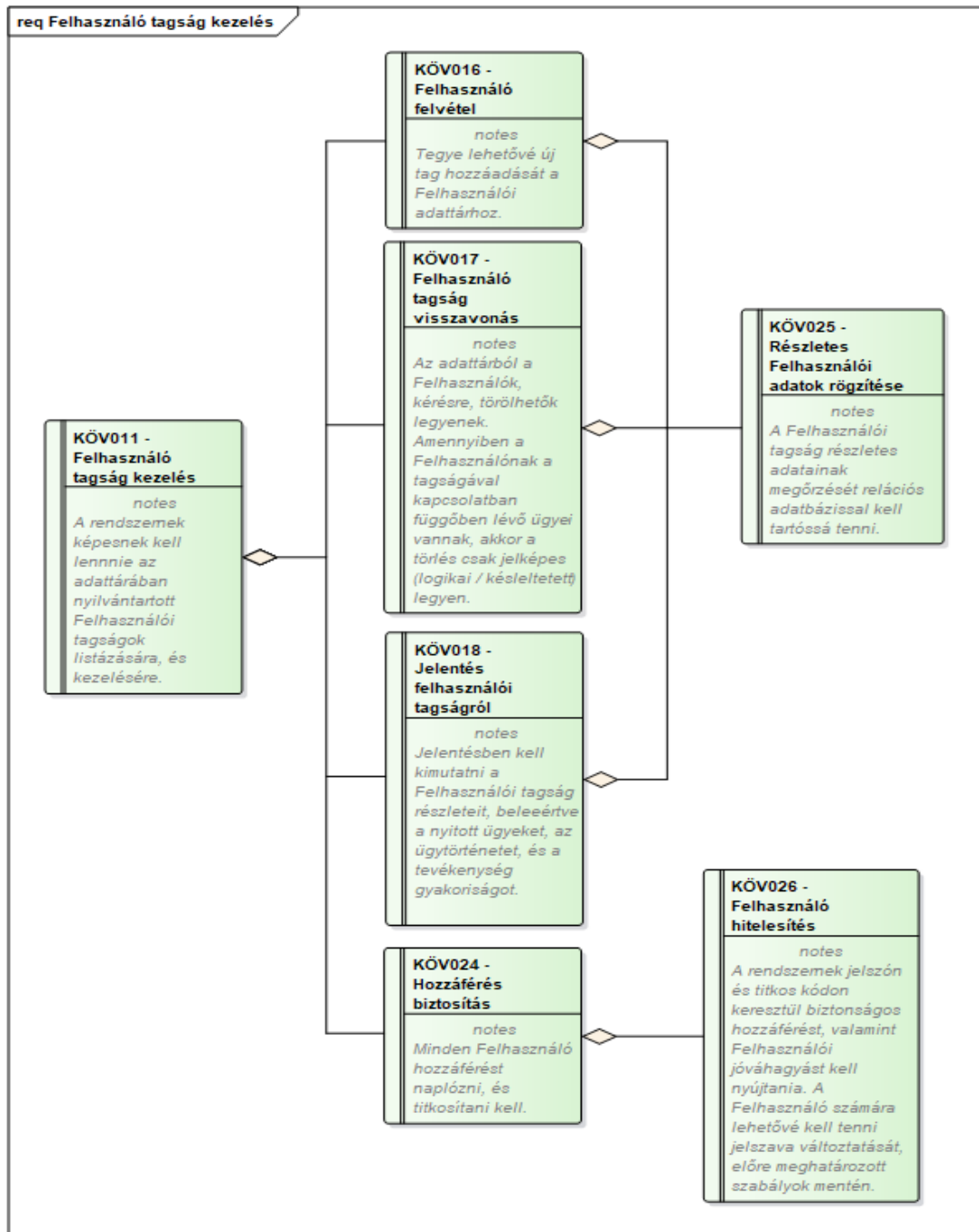
Ez a legegyszerűbb, írásos forma, amelynél a Követelmények jól szerkesztett, tömören fogalmazott, és egyedi azonosítással ellátott leírások. Elrendezésük akkor optimális, ha kategorizáltak, és témakörök mentén hierarchizáltak, vagyis származtatásuk (kibontásuk), és függőségeik (vonatkozási környezetük) formalizáltan szerkesztettek, ami biztosítja bővíthetőségüket, megvilágítja összetettségüket, valamint biztosítja a megvalósításokban a jól követhető felhasználásukat.

Írott változatuknak jól elrendezettnek (belső logikájukat tükrözőnek), tételesen hivatkozottnak (pl.: sorszámozás), és a Funkcionális követelmények esetében pontos mérőszámokkal indokoltnak kell lennie.

Grafikus modellje vizuális elemekkel segítheti a kategorizálás (pl.: színkódolás), és a függés (UML entitás kapcsolat típusok, pl.: tartalmazó függés, hivatkozó függés stb.) megjelenítését. A grafikus modellhez is kapcsolódnia kell a definíciós leírásoknak, amelyek rögzítési lehetősége grafikus tervező eszköz használata esetén eleve adott.

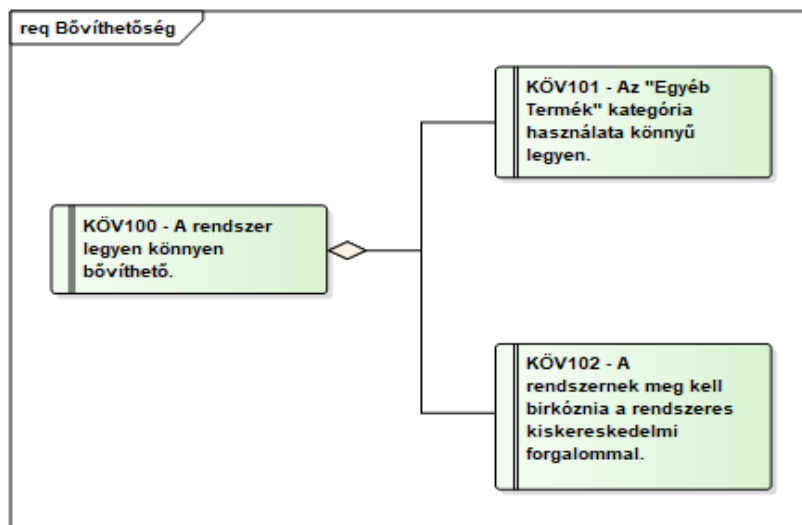
Az informatikai rendszerek műszaki termékek, ezért mérhető műszaki jellemzők mentén le is írhatók. Azonban a Követelmények nem mindegyike jellemezhető mérőszámokkal, ezért alapvetően két kategóriába sorolhatók:

- Funkcionális követelmények, amelyek tulajdonságai mérhetőek, így teljesülésük egyszerűen igazolható.



5. ábra: Funkcionális követelmény modell

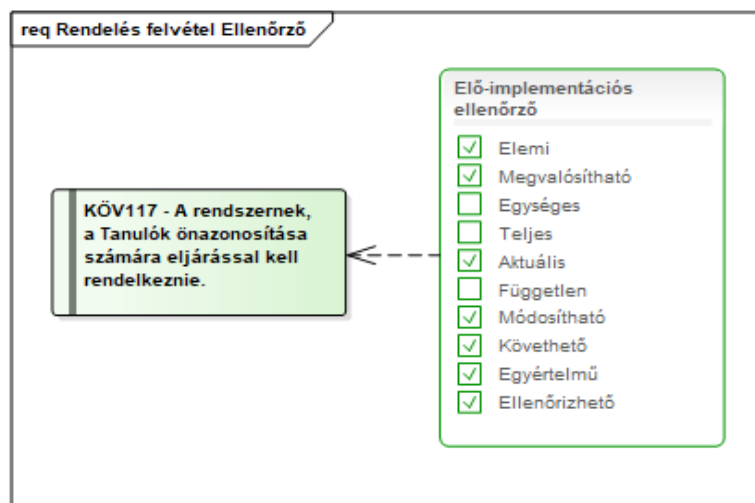
- Nem funkcionális követelmények, amik szubjektív elvárások, ezért teljesülésük nem határozhatja meg alapvetően egy rendszer elfogadását.



6 ábra: Nem Funkcionális követelmény modell

3.2.1.1.3. Korszerű megjelenítés

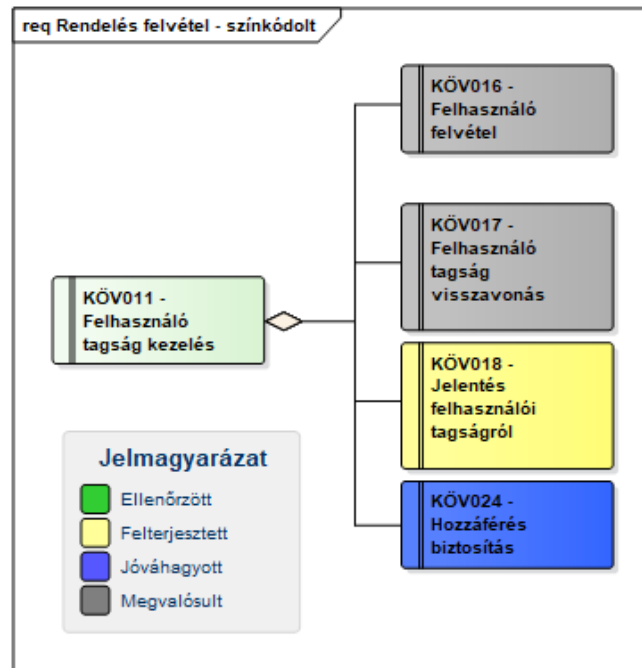
Az alábbiak példákat mutatnak fejlett Követelmény modellezésre:



7. ábra: Követelményminősítés modell

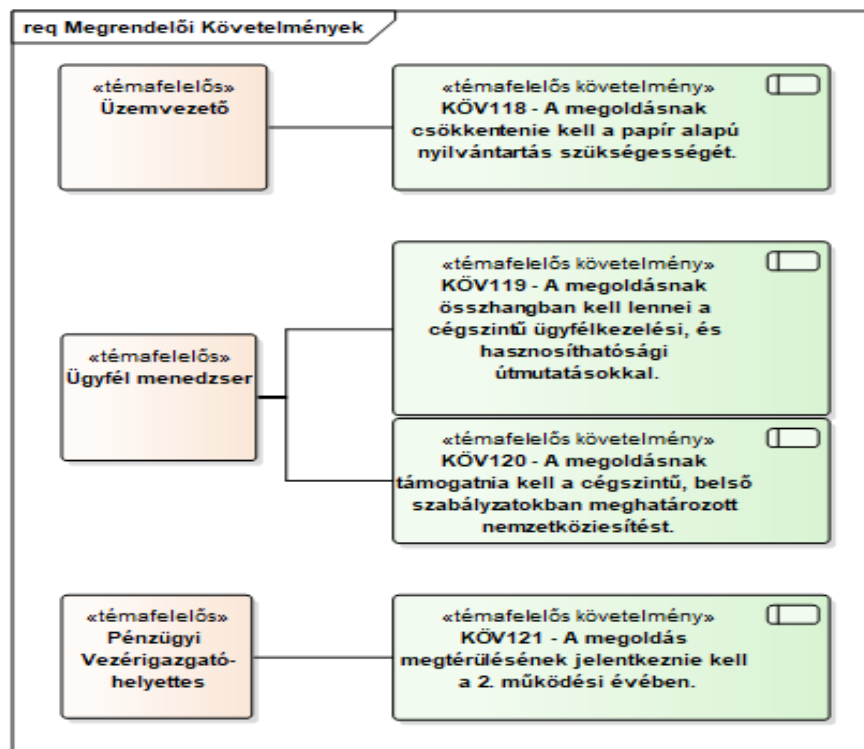
A fenti elgondolás segíti a Követelmények felhasználhatóságának, illetve feldolgozási állapotának gyors minősítését, azonosítását.

A jól megválasztott színkódolás megkönnyíti a Követelmény felhasználási/feldolgozottsági állapotának figyelését:



8. ábra: Követelmények színkódolása

A Követelmény gazdája / forrása is feltüntethető az ábrázolásban:



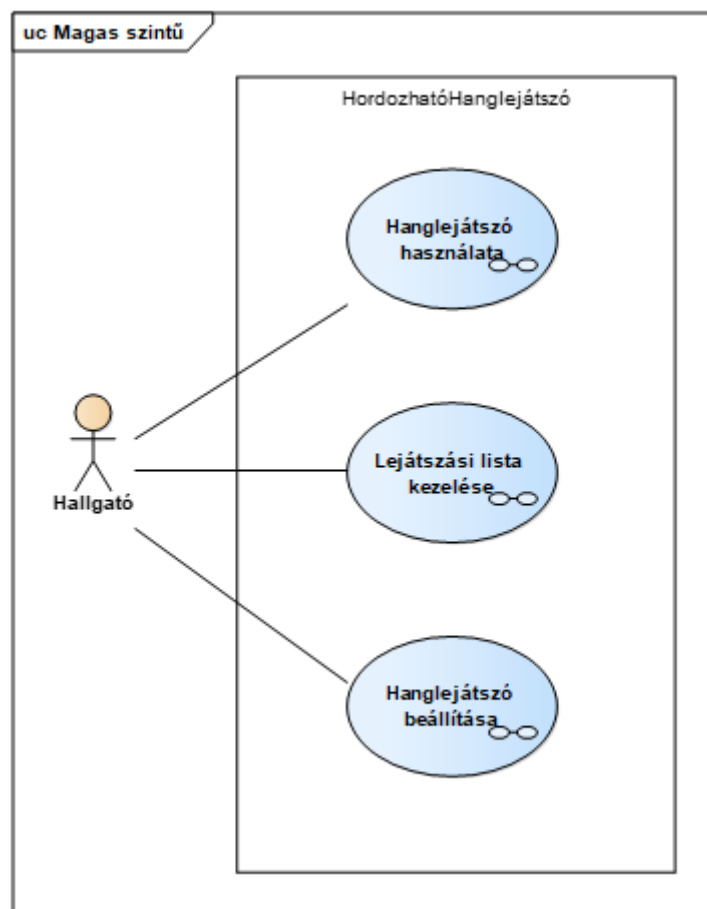
9. ábra: Követelmények származási modell

3.2.1.1.4. Esettanulmány modell

Az elvárásokat a rendszerrel elvégezhető „hozzá fordulások”, tevékenységek pontos megfogalmazása alapján bontja ki, ahol a tevékenységek az igényeket (a „funkciókat”), és nem a megvalósítást jelzik. Az esettanulmányi modellek, szemben a specifikációs leírással, egy-egy tipikus rendszer-használatot mutatnak be, ezzel rögzítve a kiépülő rendszertől elvárt funkcionalitást. A modell a Felhasználói, ún. „használati esetek” segítségével grafikus módon mutat be összetett elvárásokat, és azok kölcsönös függéseit. A „Felhasználó” itt logikai entitás, például egy szerepkör (munkakör), de akár egy másik rendszer is lehet, az igények pedig a használandó funkciók.

Az alábbi példa egy képzeletbeli, hordozható hanglejátszó berendezés használati eset leírás részletein keresztül mutatja be a követelményeket.

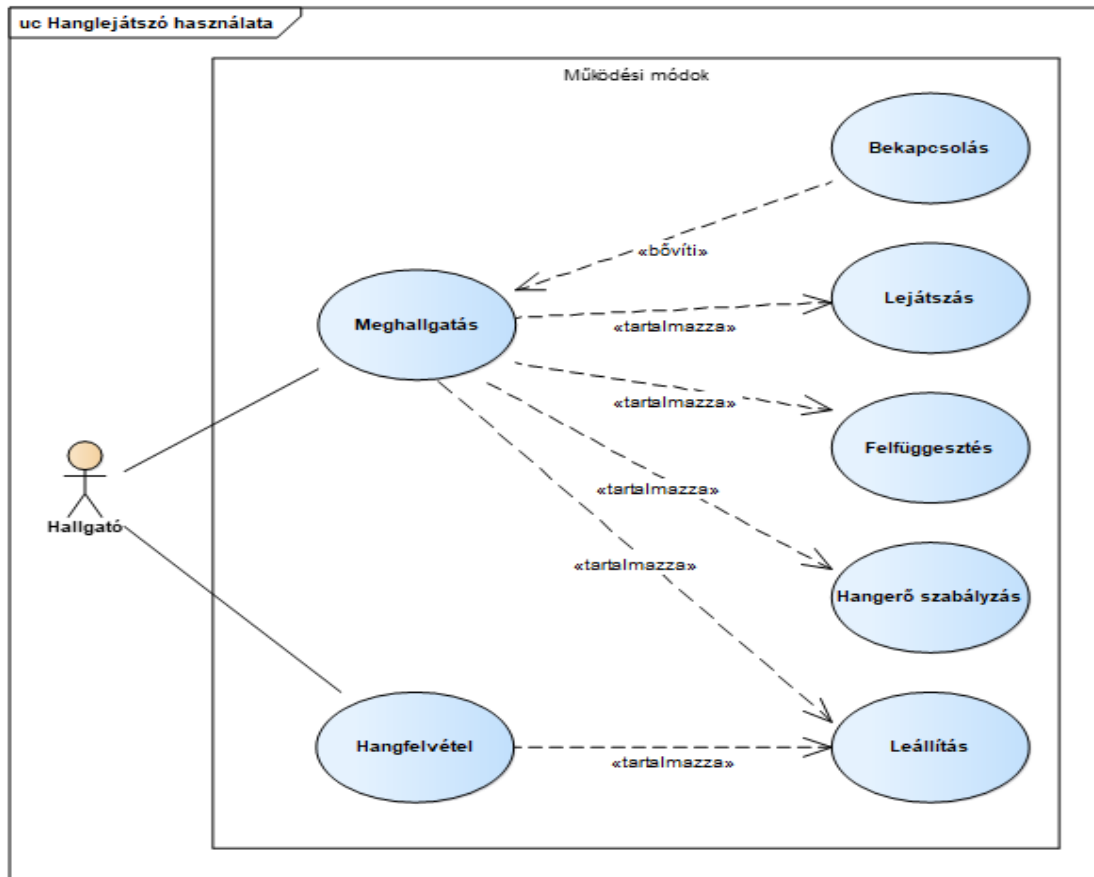
Az indító, magas szintű nézőpontot, amiből a részletezések, a „szemüveg” jellel bővített, funkcionalitásokat jelképező ovális alakzatok „mögött” elérhetők, ez az ábra mutatja be:



10. ábra: Esettanulmány modell (Use Case)

A „Hallgató” a Felhasználót, a színes oválisok az elvárt tevékenységeket, az összekötő egyenesek pedig a hovatartozást fejezik ki.

A „Hanglejátszó használata” eset részletezése:



11. ábra: Esettanulmány részletező

A szaggatott nyilak a függéseket ábrázolják.

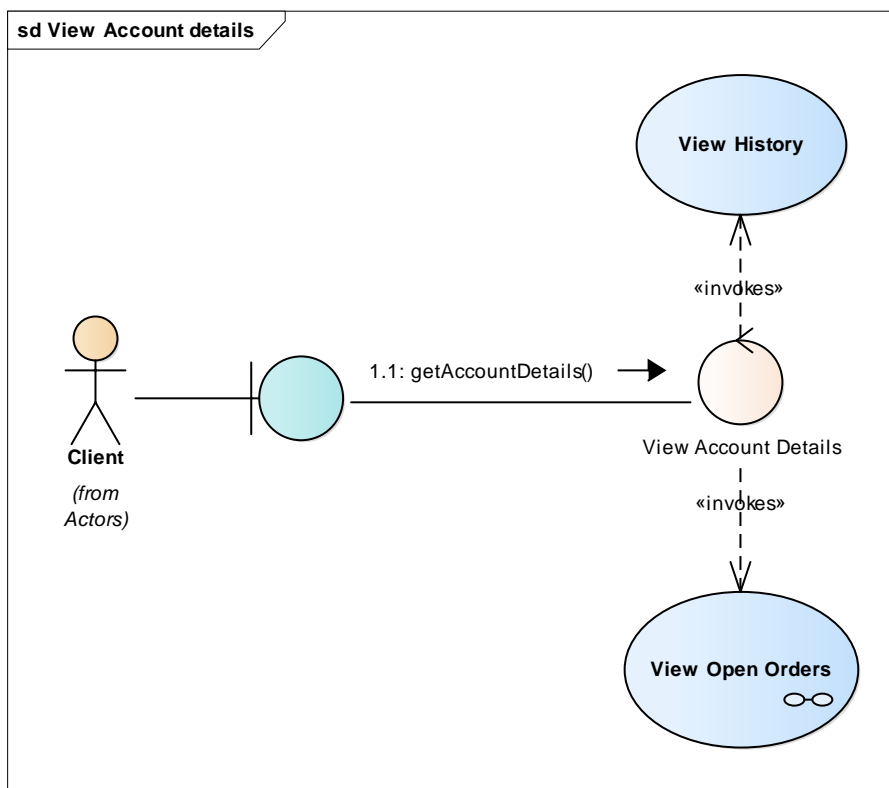
A „tartalmazás” (itt ún. sztereotípa, vagyis kategorizálás) jelentése, hogy a „Meghallgatás” funkció a másik négy tevékenységet, illetve azok használhatóságát, együtt jelenti, vagyis a „Meghallgatás” során ezek a lehetőségek, funkciók is rendelkezésre állnak, ám önállóan, a „Meghallgatás” funkciótól függetlenül nem.

A „bővítés” jelentése, hogy a „Bekapcsolás” után rendelkezésre áll a „Meghallgatás” tevékenysége, de nem szükségszerű a használata, vagyis a „Bekapcsolás” egy önálló funkció, amelynek igénybevétele után be is fejeződik a további tevékenység, ad absurdum „bekapcsolva marad” a készülék, és nem történik további használat, vagy a későbbiekben mégis csak sor kerül a „Meghallgatás” nevű funkció használatára. Az utóbbinak tehát előfeltétele az előbbi, amíg amaz független tőle.

Az ábrák (itt is!) részletes leírással egészülnek ki, amelyek rögzítik a hibamentes lefolyást, majd azok alternatív lehetőségeit, végül a várható/bekövetkező hibákat:

Név	Vásárlási tétel feljegyzés
AZ	HE_001
Leírás	A Felhasználó az eÜzlet böngészése közben megvásárolni kívánt tételt talál, és azt szeretné átmeneti listában tárolni, egy esetleges későbbi vásárlási szándékhoz előkereshető módon.
Kezdeményező	eÜzlet használó
Hasznosság	Növeli a vásárlási hatékonyságot azzal, hogy segít emlékezni a vásárlónak egy termék utáni korábbi érdeklődése alapján
Előfordulási sűrűség	A vásárlók 20%-a jegyzi fel a megvásárolni kívánt termékeit. Még a feljegyzés évében a termékek 50%-át megvásárolják.
Kiváltó ok	A Felhasználó kiválasztja a feljegyzés lehetőségét.
Előfeltételek	A Felhasználó egy terméket vesz szemügyre.
Következmények	A kiválasztott tétel megtalálható a Felhasználó feljegyzésében. A feljegyzett termék megjelölt, a Felhasználó további keresési eredményeiben, és listázásaiban.
Alapfolyamat	<ol style="list-style-type: none"> 1. A Rendszer megerősítést kér a Termék megvásárlása helyetti feljegyzésére 2. A Felhasználó megerősíti a feljegyzést (Id.: KV1) 3. A Rendszer megállapítja, hogy a Felhasználó még nem lépett be, ezért átirányítja a Belépéshez. (Id.: VZ1) 4. A Felhasználó belép (Id.: VZ2, VZ3) 5. A Rendszer eltárolja a feljegyzésre szánt tételt. (Id.: KV2) 6. A Rendszer visszavezeti a Felhasználót a mentett tételek teljes listájához.
Változatok (VZ)	<p>VZ1 A Rendszer megállapítja, hogy a Felhasználó már belépett</p> <ol style="list-style-type: none"> 1. Visszatérés az Alapfolyamat 5. lépéséhez <p>VZ2 A Felhasználó ismét kilép a Rendszerből.</p> <ol style="list-style-type: none"> 1. Visszatérés az Alapfolyamat 3. lépéséhez <p>VZ3 A Felhasználó még nem tag</p> <ol style="list-style-type: none"> 1. A Felhasználó létrehozza a tagságát. 2. A Rendszer visszaigazolja a tagság létrehozását. 3. A Felhasználót visszairányítja az Alapfolyamat 4. lépéséhez

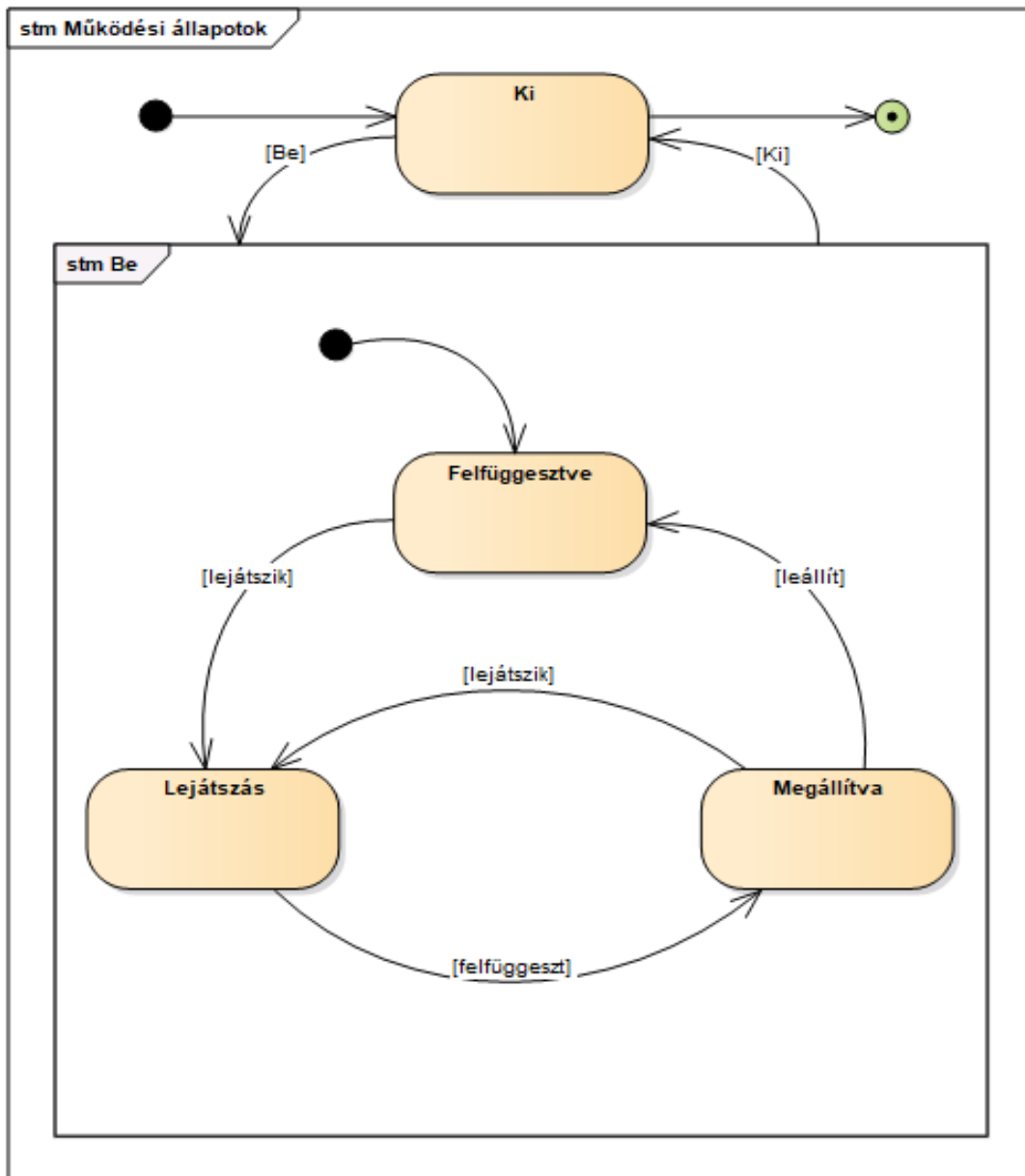
Név	Vásárlási tétel feljegyzés
AZ	HE_001
Kivételek (KV)	<p>KV1 A Felhasználó dönt az azonnali vásárlásról</p> <p>1. Lásd a „Termékvásárlás” használati esetet.</p> <p>KV2 A Rendszer nem tudja a feljegyzési listát bővíteni a tétellel</p> <p>1. A rendszer értesíti a Felhasználót a hibáról</p> <p>2. A Felhasználót visszairányítja az Alapfolyamat 1. lépéséhez</p>



12. ábra: Esettanulmány - összetett

3.2.1.1.5. Állapotgép

A rendszer a működése során megváltoztatja jellemzőit, és az őt ért interakciók hatására válasz-állapotokat vesz fel. A „működési” állapotok sokfélék, és igen bonyolult átmenetűek lehetnek, ezért önálló ábrázolásuk van, ami az előző ábra „állapot szempontú” részletezése:



13. ábra: Állapotgép

A fekete korong az állapotváltozást indító eseményt, a fekete központozású zöld korong a változás(ok) befejeződését jelenti. Az állapotok, és azonosító címkék a sárga, lekerekített sarkú négyzetek, az átmenetek lehetőségei pedig az őket összekapcsoló útvonalak görbéi. Az ábra egyszerre mutatja az átmeneteket, és azok

sorrendjét is. A rendszer működése közben felvett állapotaira vonatkozó követelmények így is modellezhetők.

Az összetett rendszerek létrehozásának alapvető lépései a koncepcióalkotás, a logikai tervezés, és a fizikai tervezés. Az informatikai rendszerek, mint összetett rendszerek terve, a Rendszerterv is ezt a legmagasabb szintű felosztást követi.

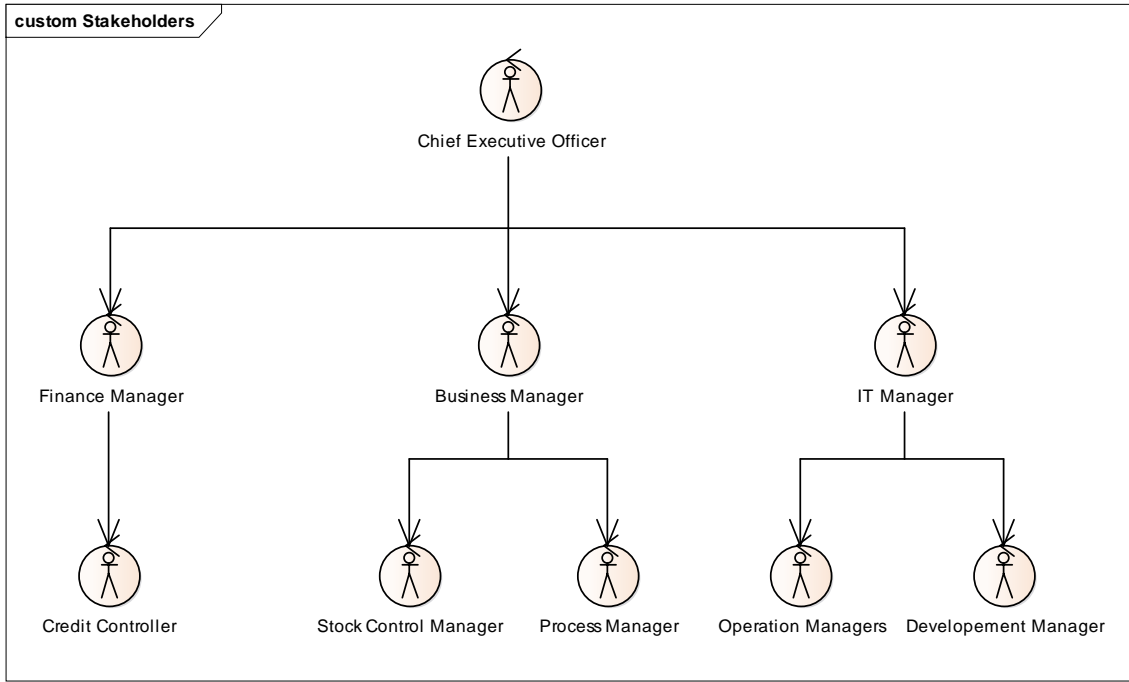
E három alapvető tervezési elem, egyben három, egymásból következő szint is, amelyben a részletek egyre pontosabb megfogalmazása a kapcsolat.

3.2.1.2. Elemzés

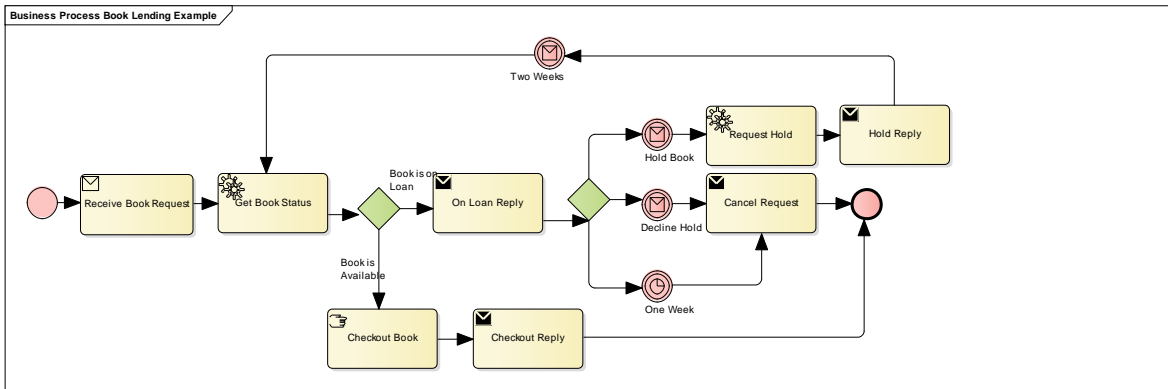
Ez a legelhanyagoltabb területe a mai informatikai megoldásoknak. A dokumentációkból alapvetően hiányoznak az "üzleti szabályok", vagy az "üzleti folyamatok" fejezetei. Habár fontosságával minden szakmabeli tisztában van, sem eszközei, sem szerepe nem kellően tisztázottak, és így hamar "túljut" rajta a legtöbb projekt.

A szemléletalakító fejezet rávilágított arra, hogy az információs rendszerek lényegi elemének tekinthető az adatok kezelése, a logikai szabályok felfedése, megformálása, összekapcsolása. A szabályok feltárása csak precíz, analizáló-szintetizáló munka eredménye lehet. A folyamatos átfogalmazás, értelmezés, elvetés, és összefésülés komoly záloga a valódi eredményességnek, a megelégedettségnek, a minőségnek.

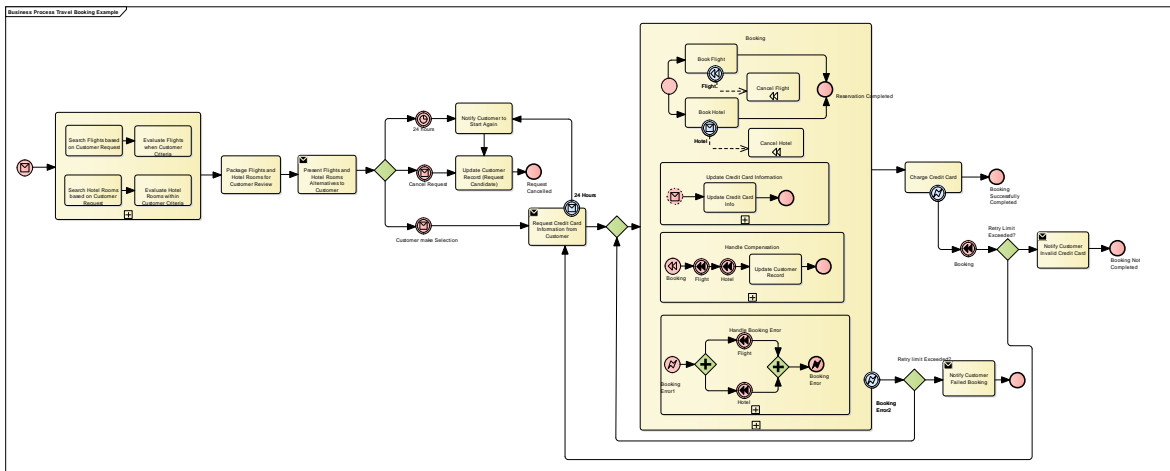
Az elemzés eredménye a magas szintű állapot leírás, modell. Az elemző munka során feltárt jellemzők alapján a tervezéskor terv változatokat lehet ezek alapján kialakítani, ezek közül pedig, a mellette, és ellene szóló érvek alapján jön létre a döntés a legalkalmasabb változat javára. A döntés megalapozottságát az elemzés minősége (követett módszere, módszeressége, alaposága, terjedelme, lényeg kiemelése stb.) határozza meg. Az elemző tevékenység teremti meg a szerves kapcsolatot az igények, és a terv(ek) között, így ennek hiánya a követelmények előírásainak, és a célkörnyezet adottságainak való megfelelés mértékét, minőségét, valamint teljességét rontják, illetve akadályozzák.



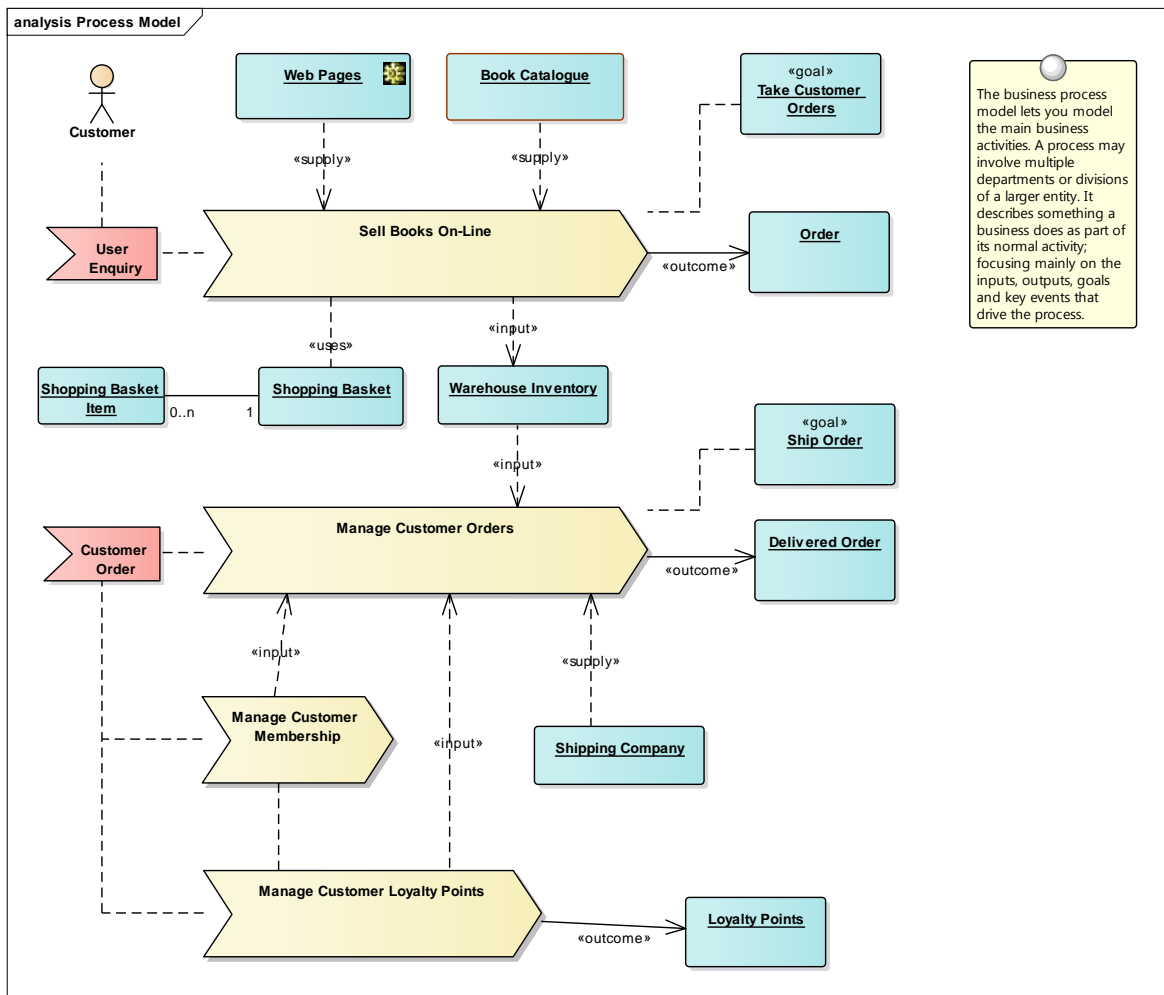
14. ábra: Projekt felelősségi modell



15. ábra: Üzleti folyamat modell – egyszerű



16. ábra: Üzleti folyamat modell – összetett



17. ábra: Megvalósítandó üzleti (cél)folyamat modell

3.2.1.3. Architektúra kialakítás

Ma, a csúcsra járatott számítástechnikai gépi kapacitás miatt, a(z) (fizikailag) elosztott rendszerek dominálnak. Alapvető architekturális elvárás a mai informatikai rendszerektől, hogy optimálisan használják fel az így rendelkezésre álló lehetőségeket.

A fizikai elosztottság mellett a rendszerek logikailag is modularizáltak, főként a hálózatos környezetet használó rendszerek. A minőség legfontosabb eredője az architektúra, aminek része az elemek kiválasztása, és az összeépítési logika. Itt a hangsúly az alkotórészek szempontjából a kiválasztásuk, mind követelmény, mind technikai/technológiai oldali indokainak megalapozott kifejtésén van, míg a működési logika nézőpontjából a rendszer elemek kiépítési módjának, a felhasznált technológiák választásának a bizonyításán.

A kellő részletességű koncepcionális elvek létezése, az egyes megoldás részletek összefüggéseinek megfogalmazása, és mindezen elképzeléseknek a követelményekből való szigorú levezetettsége a garanciája a jó tervnek, és a jó megoldásnak.

A Rendszer Architektúra az a keretrendszer, ami a rendszer határait, és fő elemeit kijelöli, az alapvető elvárásokat magas szinten teljesíteni képes működési móddal együtt. Koncepcionalitását fokozatos finomítással őrzi meg, a Logikai, és Fizikai tervezés során megjelenő problémákra válaszul adott megoldási elemek lényegi jellemzőinek visszacsatolásával.

3.2.1.4. A tervdokumentáció elvárt tartalmi egységei

Az összetett rendszerek létrehozásának alapvető lépései a koncepcióalkotás, a logikai tervezés, és a fizikai tervezés. Az informatikai rendszerek, mint összetett rendszerek terve, a Rendszerterv is ezt a legmagasabb szintű felosztást követi.

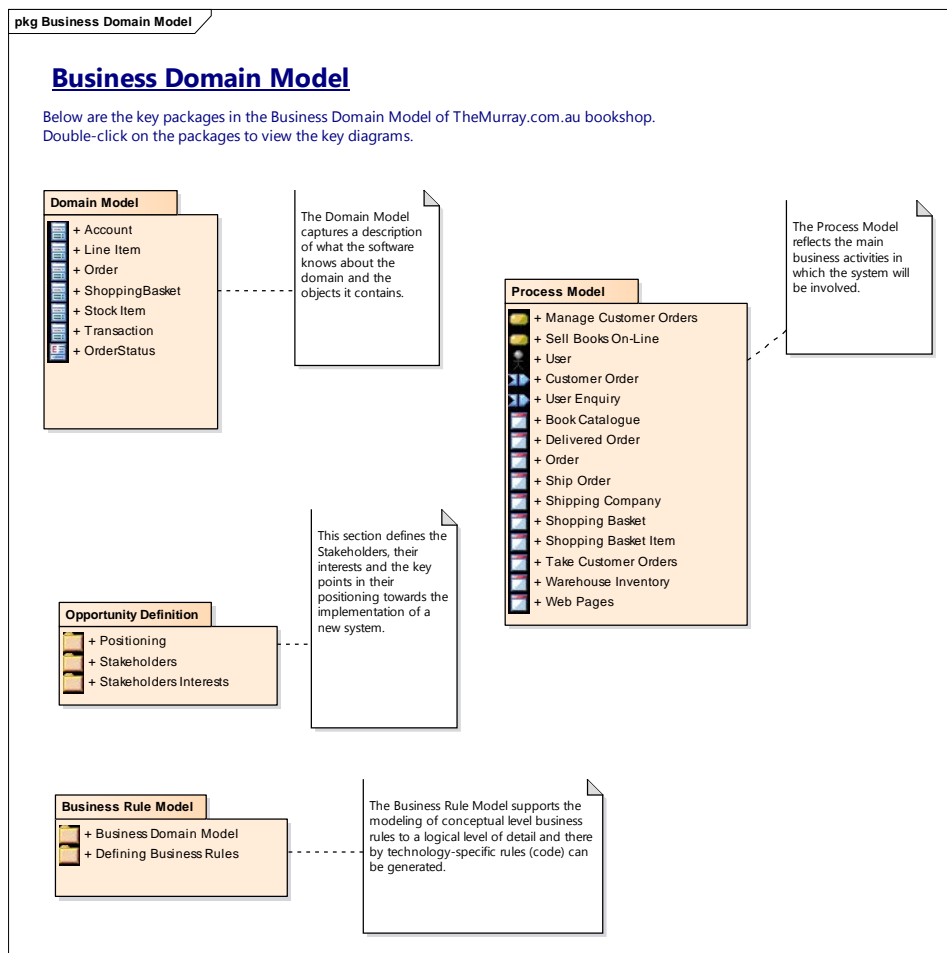
E három alapvető tervezési elem, egyben három, egymásból következő szint is, amelyben a részletek eltérő szempontú, és egyre pontosabb megfogalmazása a kapcsolat.

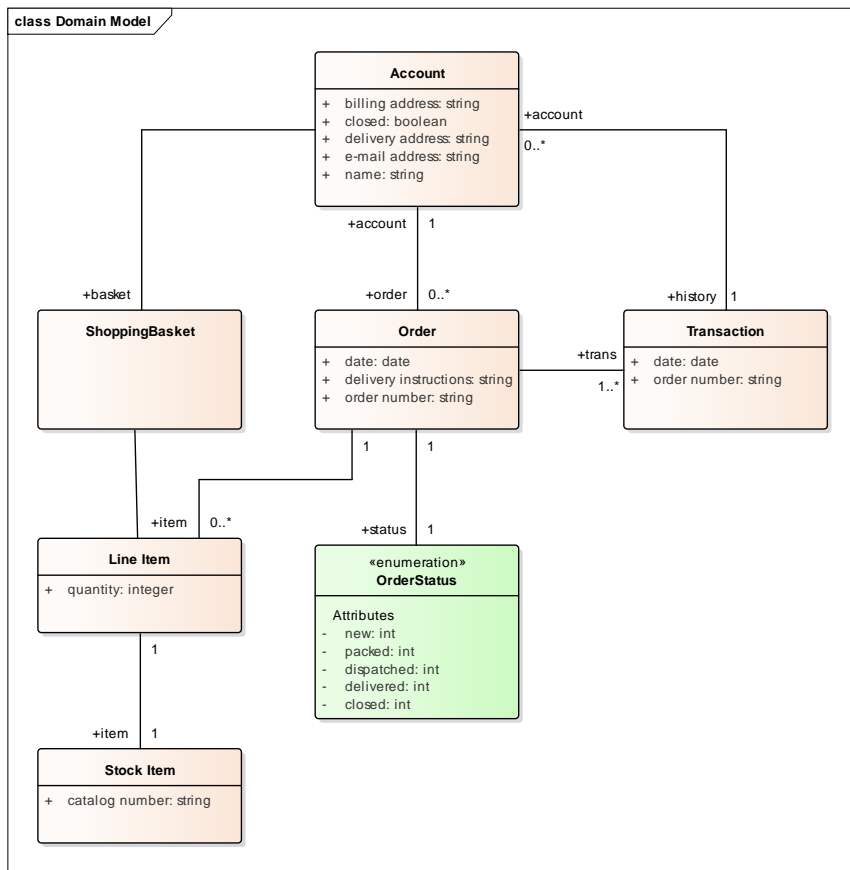
3.2.1.4.1. A koncepcionális tervezési szint

A koncepcionális tervezés alapja egy magas szintű igény, ami alapján megfogalmazhatók a fejlesztési célok, és összegyűjthetők az alapvető követelmények.

A Követelmény jegyzék a rendszerrel szemben támasztott igények, elvárások különböző mélységű és struktúrájú megfogalmazása. Tapasztalat, hogy az esetek többségében, különösen közepes, és nagy rendszereknél a követelmények megfogalmazása az informatika gyorsan változó lehetőségeinek tárháza, illetve a megrendelői oldal saját elképzeléseinek kialakulatlansága miatt nem kielégítő, emiatt a tervezett rendszer igen sokszor alapvetően egy „fekete doboz”, mind a megrendelőnek, mind a kivitelezőnek egyaránt. A pontatlanul megfogalmazott üzleti igények, és inkább tapogatózó, mint letisztult követelmények alapján a koncepció is igen sokszor elnagyolt. Emiatt megfordul a folyamat természetes iránya, és a koncepció alkotás során folytatódik a követelmények összegyűjtése, és rendszerezése. A követelményekre csak úgy lehet alapozni, ha rendezettek. A strukturálással, osztályozással gyorsan észlelhetők az egyenszilárdsági, a pontatlansági, a részletezettségi problémák. Ma már, vizuális modellezési támogatás nélkül, sem a követelmények kezelése, sem a koncepció ismertetése, elfogadtatása nem lehet sikeres.

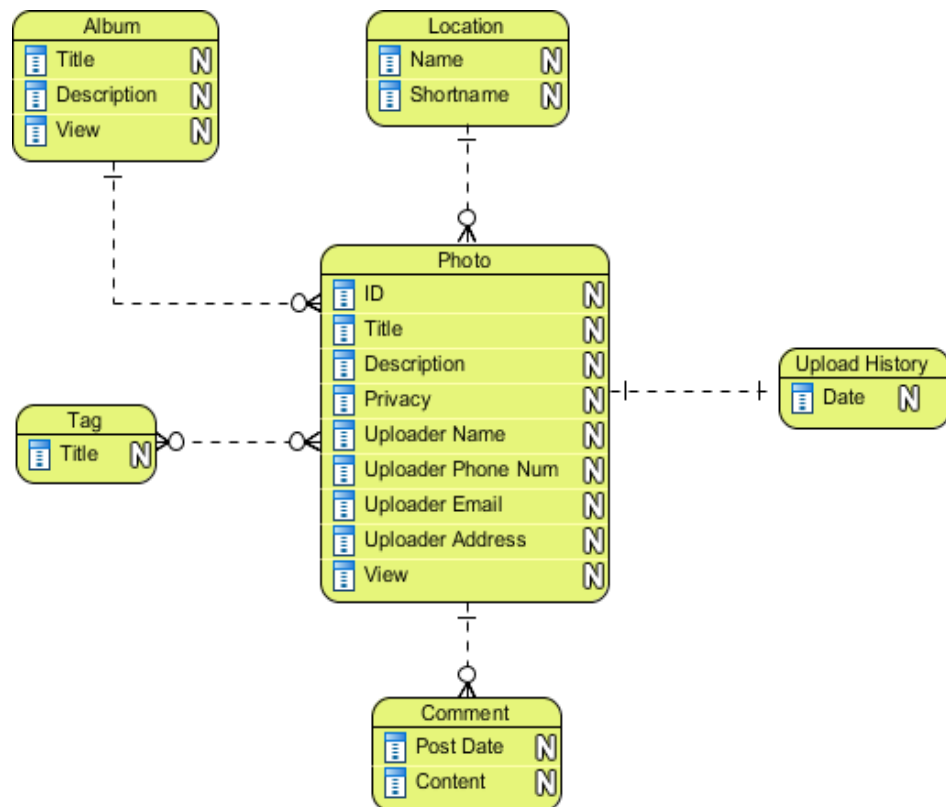
A Konceptió a rendszer legfontosabb szolgáltatásaira koncentrál, ezzel átfogó, érzékletes képet mutat be a megrendelőnek. Része az ún. „Szakterületi modell” (“Domain” modell), ami a tervezett informatikai megoldás befogadó környezetét írja le, önmagában, tehát "As Is" (ahogy van). Ez a környezet az, amibe a követelményeknek, igényeknek megfelelő informatikai támogatást kell beilleszteni.





18. ábra: [Fogalmi-, Domain-,] Szakterületi modellek (konceptcionális)

A Konceptcionális modell alapja a tervezett, támogató rendszerrel szembeni elvárások rögzítése, amik pontosan megmutatják a támogatás mibenlétét. A modell a támogató környezet fogalmait használja a leíráshoz, és nem jelenik meg benne informatikai meg gondolás, sem semmilyen megoldási kezdemény, ezért Fogalmi modellnek is nevezik. Hiánya egyértelműen súlyos hiba, mert nincs meg a viszonyítási pont, amihez az informatikai megoldás csatlakozik, valamint nincs meg a „bizonyítéka” annak, hogy a kivitelező megértette a leendő Rendszer szerepét, és helyét a befogadó környezetben, valamint a Megrendelői munkafolyamatokat, vagyis a Megrendelő „üzletét”.



19. ábra: Fogalmi (konceptcionális) adatmodell

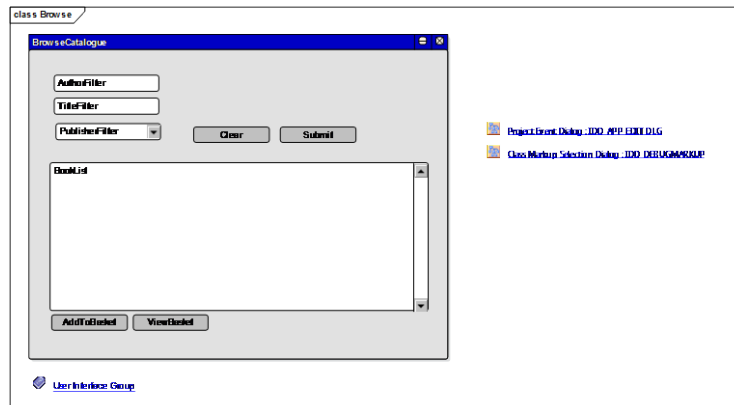
A Követelmények megítélésének szempontjai közül a fontosak:

- Fogalom szótár kialakítása, és következetes (!) használata.
- Pontos követelmény azonosítási rendszer (egyedi kódolás).
- Követelmények hierarchizálása, osztályozása.
- Tendenciózus ábra használat.
- Érthető, logikus tagolás.
- A funkcionális, és nem funkcionális követelmények szétválasztott megjelenítése.
- A felelős szerepkörök feltüntetése.

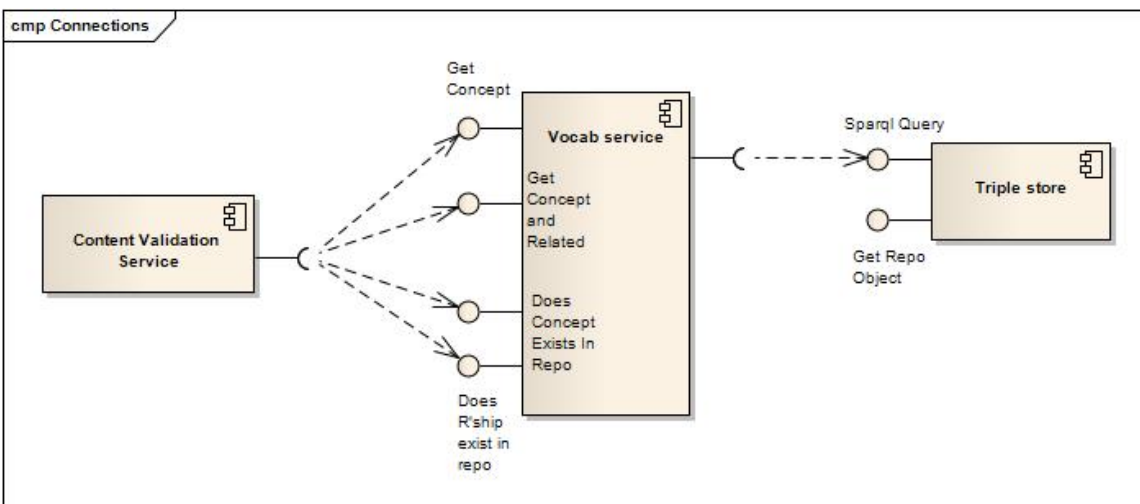
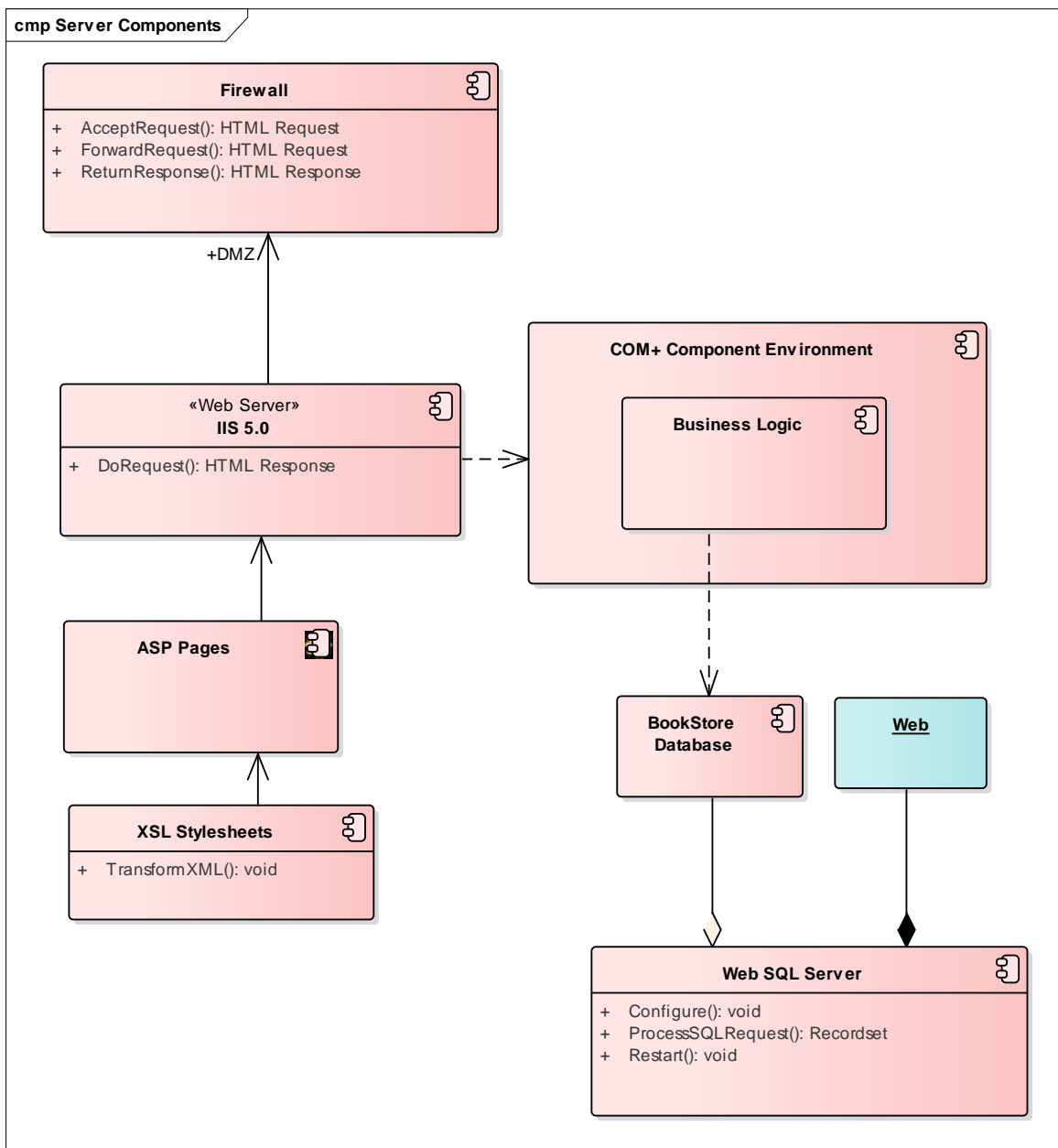
3.2.1.4.2. A logikai tervezési szint

A következő szint a Logikai terv, ahol már belép az informatika, és itt jelentkezik a határ a szakterületi nézőpont és az informatikai nézőpont között, aminek áthidalása komoly szakmai kihívás minden projekt számára. Ennek a szintnek a terve bemutatja a rendszert támogató adatokat és funkciókat, de nem tér ki arra, hogy részleteiben hogyan működnek ezek az elemek. Ez utóbbi jellemzők a Fizikai tervben jelennek meg, és ez jelenti a konkrét megoldást is.

Egy Konceptcionális modellből számos Logikai terv, és ebből számos Fizikai implementáció hozható létre. A sokféle lehetőség létezése miatt elengedhetetlen része a tervezési szinteknek az a "levezetés" (konceptió) magyarázat, ami bemutatja egy adott szint tervének, az előző szint elemeiből való származtatási szabályait. Ha nem található, vagy nem könnyen érthető ez a szabály halmaz, akkor a tervezés ad-hoc jellegű, és mint ilyen, nem konzisztens, így végső soron nem megfelelő.

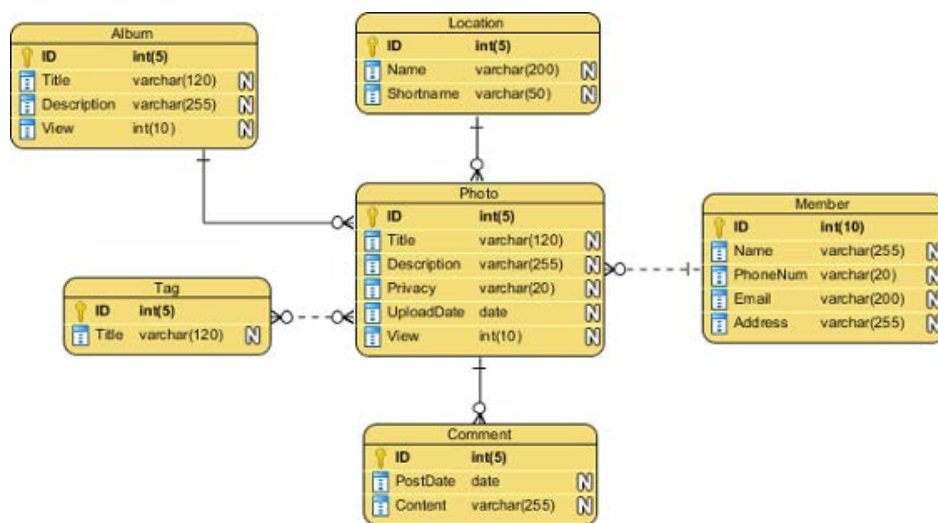


20. ábra: Felhasználói felület (User Interface - UI) Logikai modell



21. ábra: Logikai összetevő (component) modellek

Alapvető fontosságú, hogy a logikai terv nem tartalmazhat konkrét megoldási elemet, jellemzőt, ami egy konkrét technológiához, technikához köthető. A Logikai terv lényege éppen az, hogy a technológiai / technikai lehetőségek (a megvalósításhoz alkalmazott eszközök, termékek) nem befolyásolják a természetét, mert lényege éppen az, hogy bemutassa, mit kell tudnia a rendszernek, és nem azt, hogy hogyan. Az adatok modellezésénél, vagyis itt a Logikai Adatmodellen ellenőrizhető a legjobban ennek a „semlegességnek” a jelenléte: amennyiben például Relációs modelltől van szó, úgy a modell nem tartalmazhat ún. Idegen Kulcsokat (Foreign Key), és (még mindig!) tartalmazhat N:M kardinalitású (számosságú) kapcsolatokat.



22. ábra: Logikai adatmodell

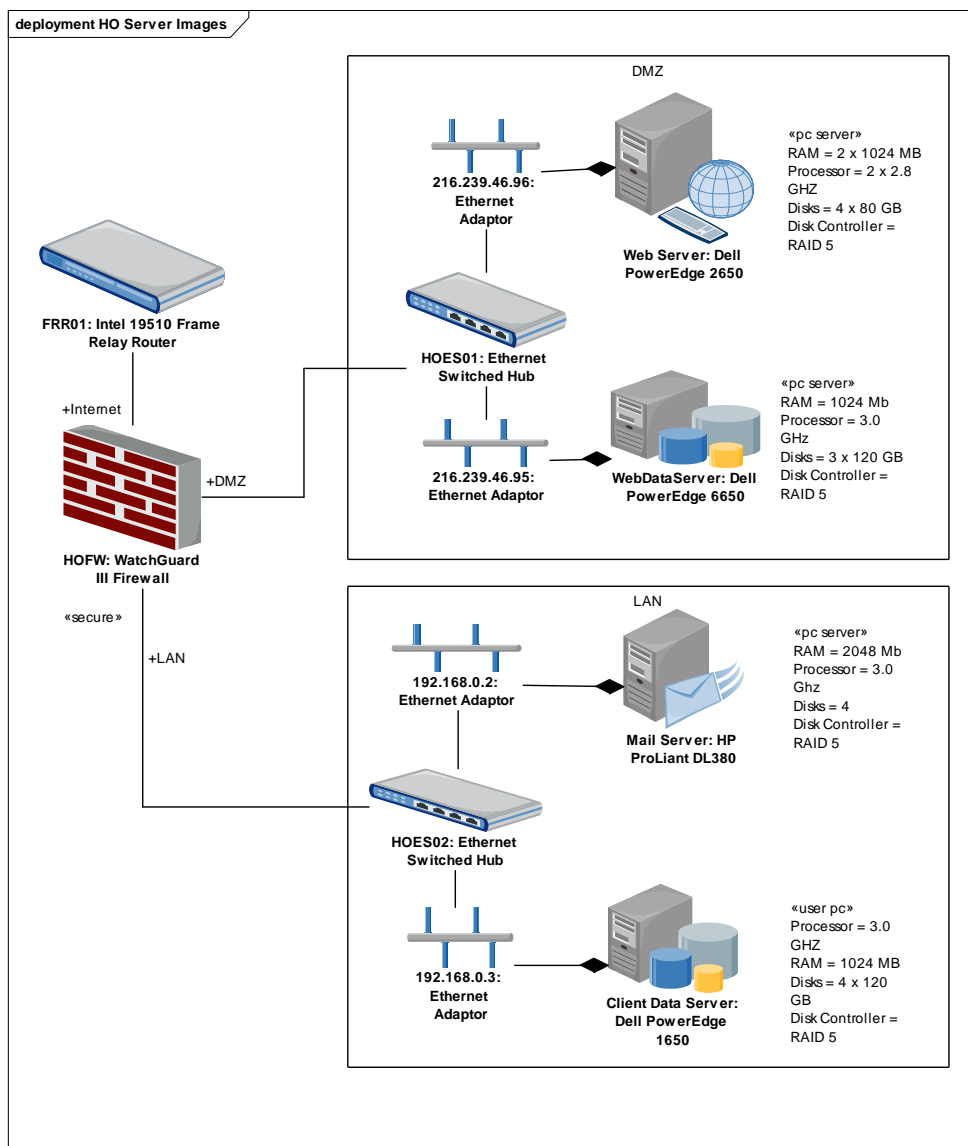
A Logikai terv megítélésének szempontjai közül a fontosak:

- A terv kialakítási szabályainak, döntések útján keletkező elképzeléseinek kifejtése, indoklással.
- Nem tartalmazhat válaszokat, megoldásokat a „hogyan” kérdésre.
- A Követelmény kapcsolatok pontos kimutatása.
- Tendenciózus ábra használat.
- Érthető, logikus tagolás: fő-, és al folyamatok, logikai adatmodell.
- A megoldásban érintett, felelős szerepkörök feltüntetésé.
- A terv megrendelői elfogadási nyilatkozattal zárul, amelynek hiánya súlyos minőségi probléma.

3.2.1.4.3. A fizikai tervezési szint

Ezen a szinten már minden terv elemnek egy konkrét megoldást kell bemutatnia, így a korábban készített modellek itt átalakulnak, és kiegészülnek a választott

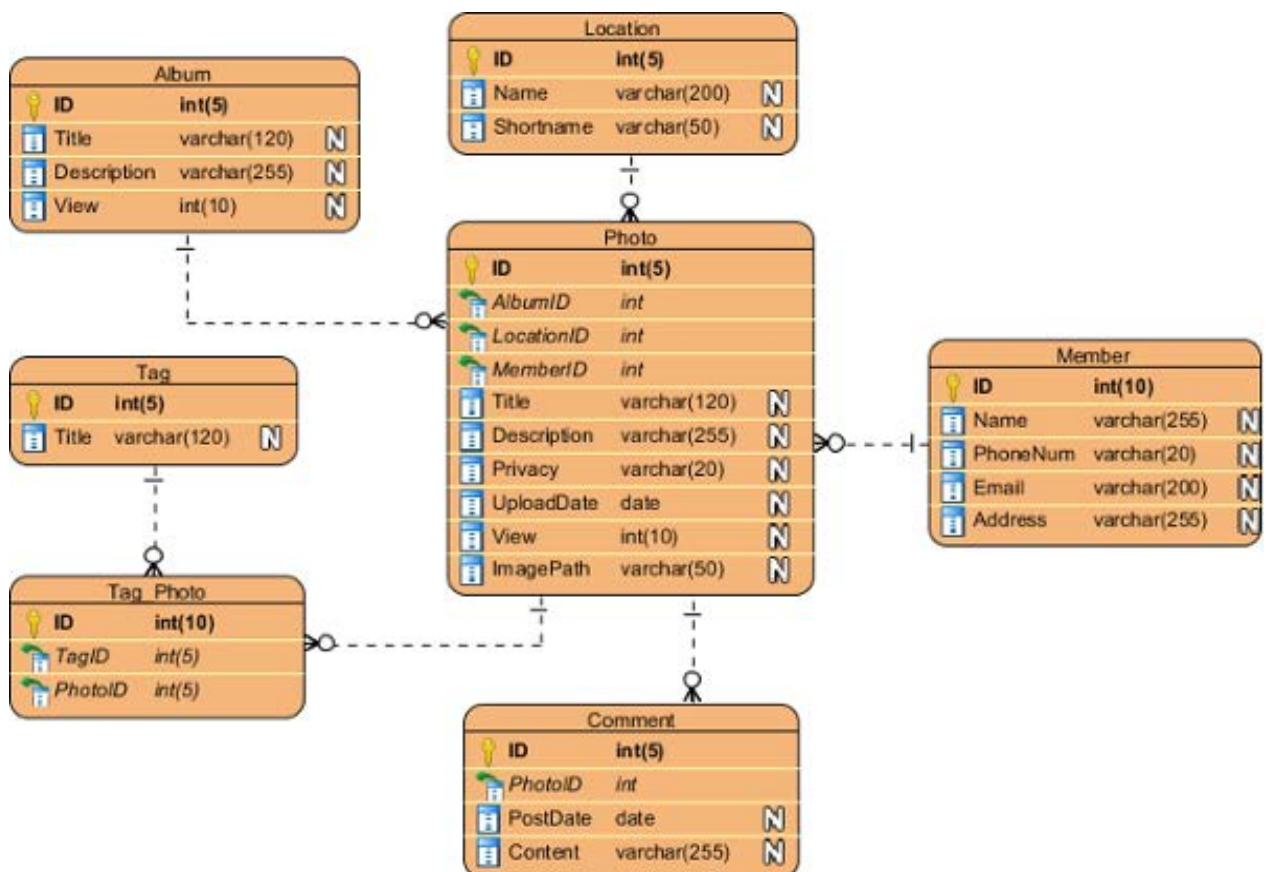
technológiák nyújtotta lehetőségek, és korlátok következményeivel. Itt érhető igazából tetten az, hogy a logikai tervből számos fizikai terv, implementáció készülhet, hiszen a választott technológiák függvénye a terv (immár konkrét, kézzelfogható) tartalma. Itt is, sőt, itt csak igazán (!) kiemelten fontos a korábbi szintből történő levezetés, indoklás, hiszen itt akár jelentősen is kiegészülhet a logikai („tiszta”) terv olyan számosságú „segédlettel”, ami mind az anyagi, mind a humán, mind az idő ráfordítást nagyságrendekkel módosíthatja (akár a kellemes, „olcsóbb” irányba is, habár erre nehéz lenne példát találni). Mivel itt elérkezik a megoldás a valódi természetéhez, a későbbi üzemeltetéssel kapcsolatos megoldásokat is itt már jelezni kell!



23. ábra: Hálózati Fizikai modell

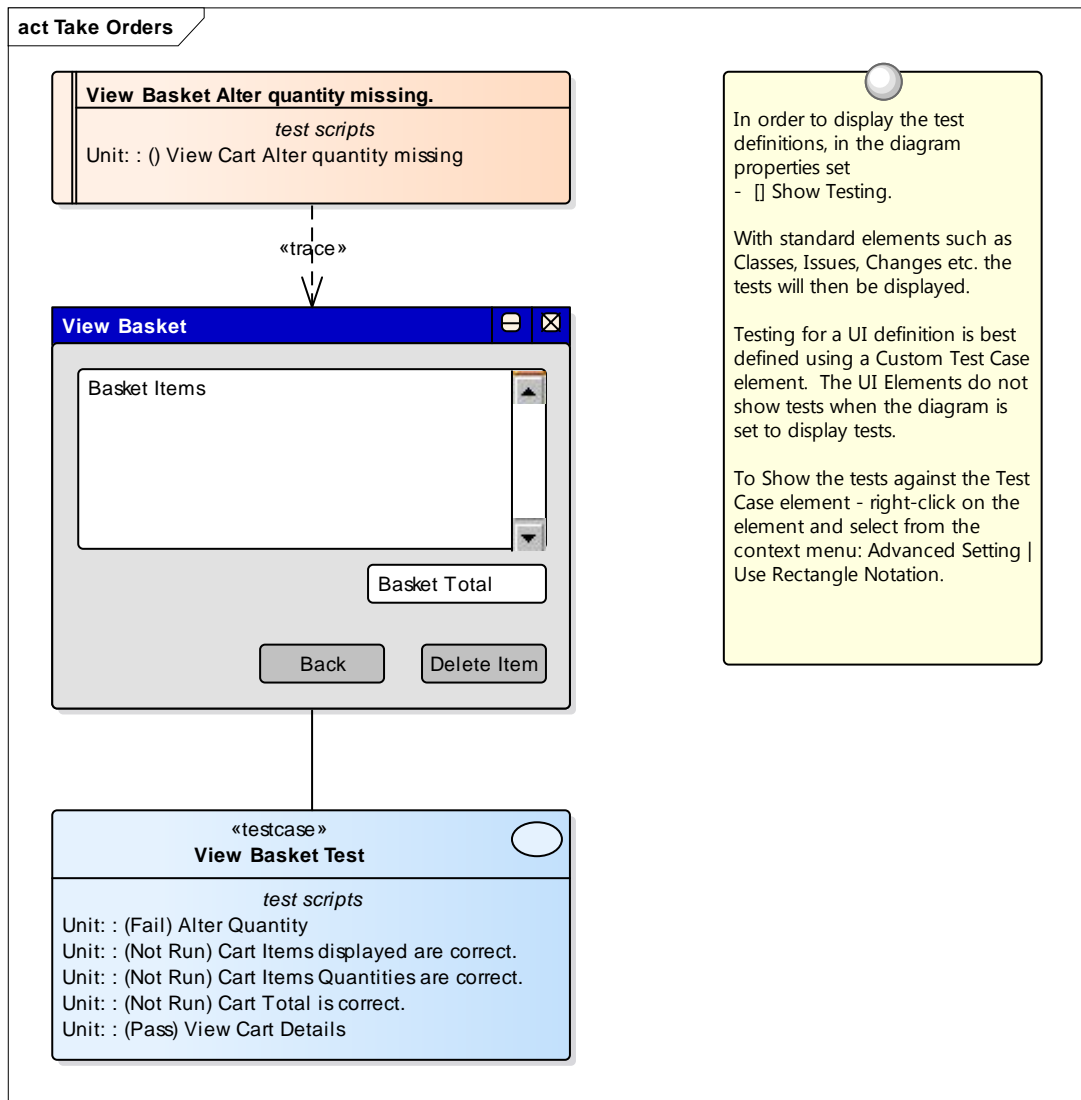
A Fizikai terv megítélésének szempontjai közül a fontosak:

- A terv kialakítási szabályainak, döntések útján keletkező elképzeléseinek kifejtése, indoklással.
- Válaszokat, megoldásokat KELL tartalmaznia a „hogyan” kérdésre.
- A Követelmény kapcsolatok pontos kimutatása.
- Tendenciózus ábra használat.
- Érthető, logikus tagolás: fő-, és alflowamatok, interfészek, alkalmazott technológiák, technikák, megoldás támogatói elemek, készletek, keretrendszerek, eljárások; fizikai adatmodell.
- A megoldás üzemeltetésében érintett, felelős szerepkörök, elvárt támogatói munkafolyamatok, karbantartási, beállítási tevékenységek feltüntetése.
- A terv a megvalósulása után átadás/átvételi nyilatkozattal zárul, amelynek hiánya súlyos jogi probléma.



24. ábra: Fizikai adatmodell

Természetesen a tesztelésre vonatkozó előírások, elképzelések is ugyanilyen fontosságúak, de mindemellett a kielégítő tervezésnek éppen az az egyik fő velejárója, hogy a teszteléshez szükséges információk szinte „kihullnak” belőle.

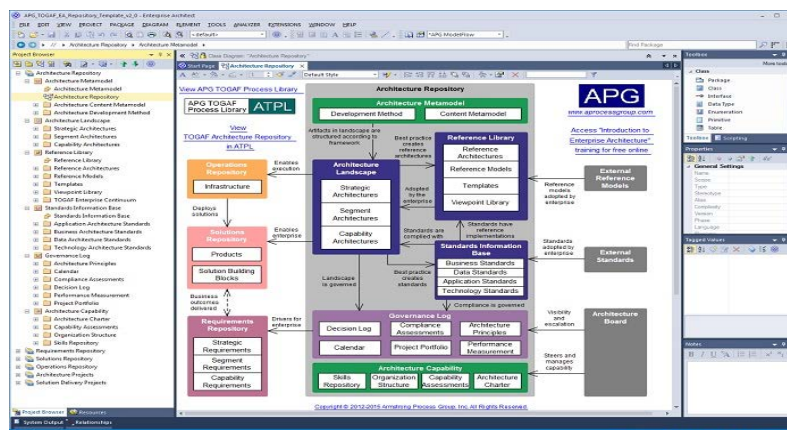


25. ábra: Teszt terv

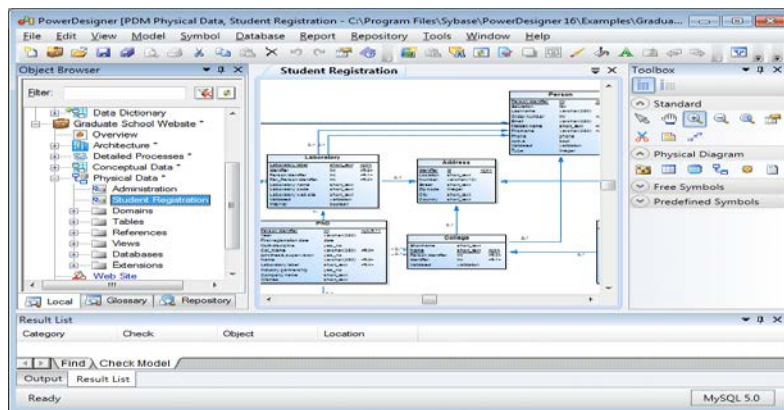
3.2.1.5. Tervezés támogatása

Egy közepes méretű informatikai projekt is több száz tervezési objektumot állít elő. Ezek követése, összefüggéseik számon tartása, a napi munkavégzés hatékonysága, az eredmények egységes, és sokrétű publikálása csak rossz minőségben végezhető el, ha papír alapú, vagy nem kifejezetten informatikai fejlesztési célokat szolgáló felhasználói alkalmazások (pl. MS Excel, MS Word, MS Visio stb.) segítségével történik a kivitelezésük. Kijelenthető, hogy megfelelő minőségű eredményt csak korszerű, a célnak legjobban megfelelő eszközökkel lehet elérni.

Az informatikai piac ma már rendelkezik a fejlesztési életciklus egészét támogató tervező, és alapkód generáló eszközökkel, amelyek szerteágazó funkcionalitásukkal hathatósan támogatják például a terv változások követését (hatáselemzés), a változatok elkülönítését, az iparági sztenderd modelleket, szabványokat, és mindezek sokrétű vizuális megjelenítését, testre szabható, sablon alapú dokumentálását, és nem utolsósorban a közösségi ("csapat") munka lehetőségeit, alapértelmezetten (!) projekt szerű keretek között. A piac jelenlegi két vezető terméke a Sparx Systems cég „Enterprise Architect”, és a SyBase cég (ma már SAP) „Power Designer” elnevezésű eszközei.



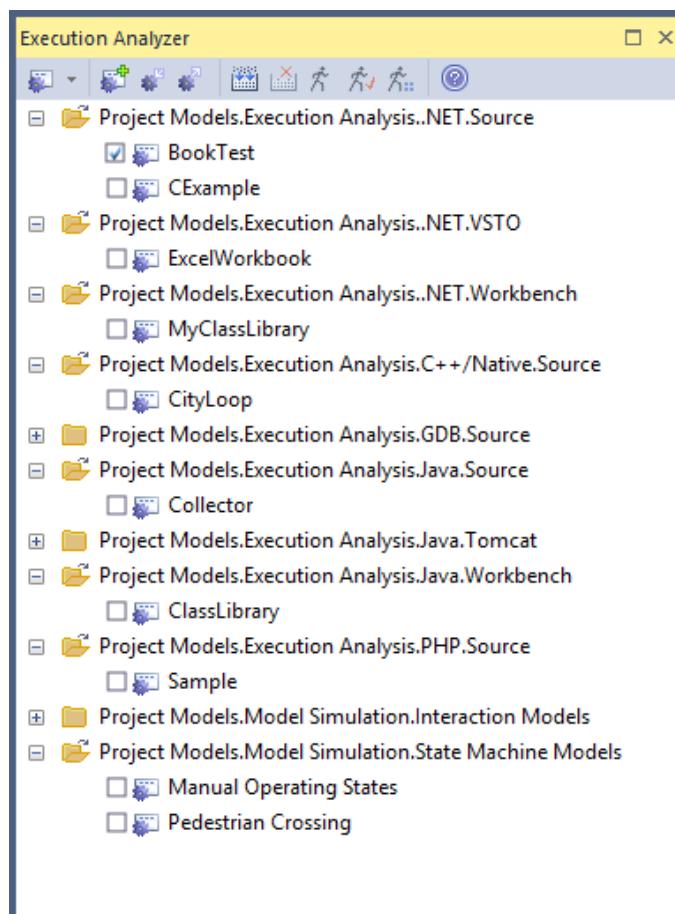
26. ábra: A Sparx Systems „Enterprise Architect” tervező eszköze



27. ábra: A Sybase "Power Designer" tervező eszköze

Tervezés támogató képességeik közül a legfontosabbak:

- A rendszerelemek és modellek, tervek és kapcsolataik változatos elrendezésű tárolása.
- A rendszert leíró különböző dokumentum típusok hivatkozási, vagy teljes befogadású tárolása.
- A tárolt rendszer adatokból, a rendszert leíró dokumentumok automatikus, változtatható tartalmú, formátumú és típusú előállítás.
- A terv változatok követése.
- Nemzetközi, és legjobb gyakorlat szabványok, minták alkalmazási lehetősége.
- Kódgenerálás, és visszafejtés.
- Adatszerkezet generálás és visszafejtés.
- Termék installációs csomag generálása.
- Kódváltozás követés.
- Egyéni munkakörnyezeti perspektíva alkalmazása.
- Folyamatelemzés, szimuláció, navigáció.



28. ábra: Folyamatelemző (Sparx: Enterprise Architect)

3.2.1.6. Szemléltetés

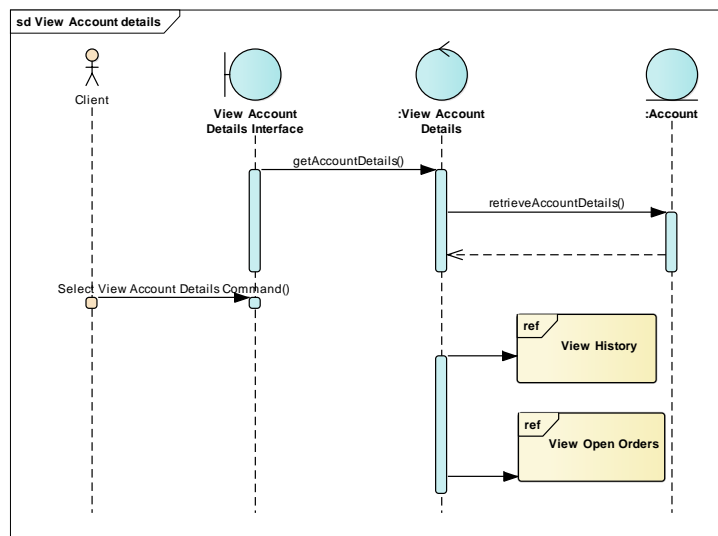
Igen gyakori megoldásnak számít a hazai gyakorlatban a “kilószámra” létrehozott, terjengős, szerkesztetlen, összefüggéstelen dokumentálása a rendszer egészének, vagy részeinek, mindenféle fórumon, és bármiféle hallgatóságnak.

Ahogy a Zachman ontológia is bemutatja, a célközönség, a “Kinek szól?” megszabja azt a perspektívát, kontextust, fogalmi környezetet, amelynek terében az informálás valóban létrejöhet. Felesleges technikai részletekkel és informatikai szakzsargonnal bombázni egy rendszer bemutatásakor a vezetői szintet, és többnyire eredménytelen vállalkozás is. Főként az írott szöveg túlsúlya jellemzi a mai dokumentálási elképzeléseket.

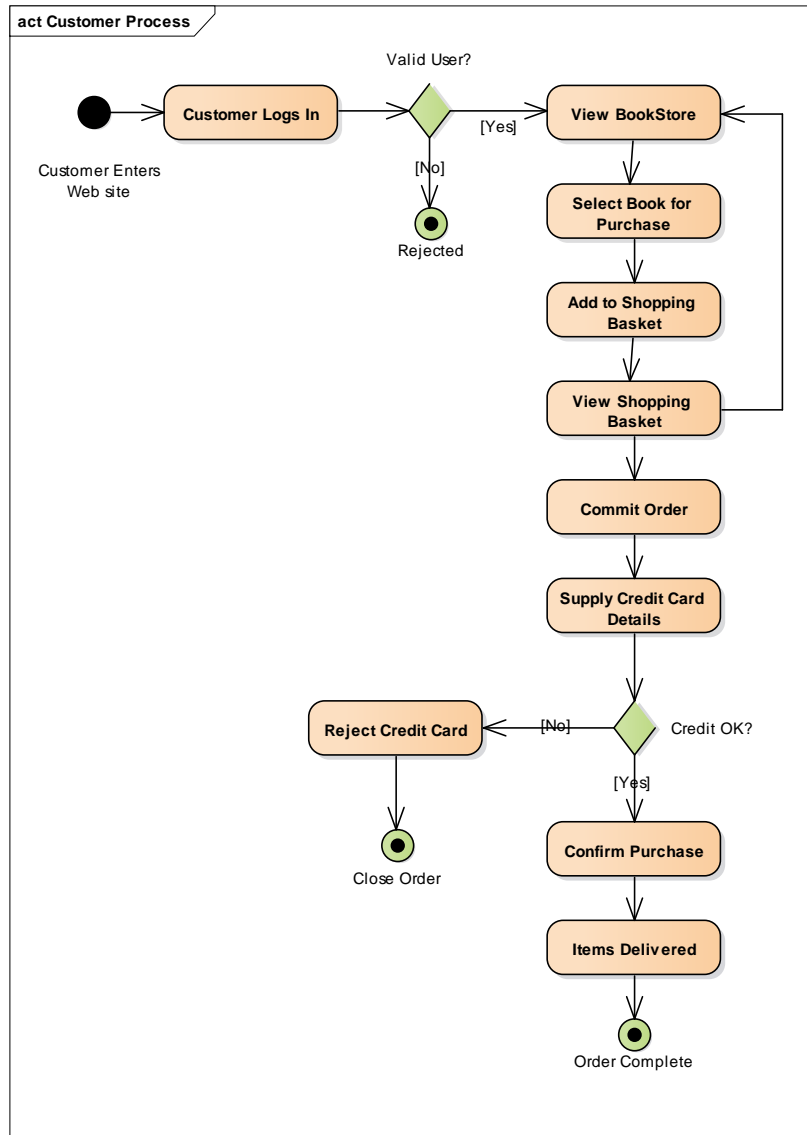
Véleményünk szerint az informatikai rendszerek természetéből - ami az adatok rögzítésével párhuzamosan a kezelésükkel, értelmezésükkel kapcsolatos logikai szabályok felhalmozását, és ezeknek többféle cél elérése érdekében összefüggésekbe rendezését jelenti - fakadóan a leíró részek mellett kiemelt hangsúlyt kell kapnia az ábrázolásnak, a könnyen érthető megjelenítésnek.

Ahogy a mérnöki területeknek megvannak a szabványos jelölés rendszerei, úgy létezik az informatika területén is az UML jelölő nyelv, amivel szabványos ábrázolás valósítható meg. Maga a nyelv is definiál ábrázolási szisztémákat, ábra típusokat, amelyek összhangba hozhatók a Zachman nézetekkel. Az ábrázolást “kiegészítő”, “díszítő” elemként használó felfogás már idejét múlt elképzelés, és úgy az informatikai rendszerek résztvevői közötti kommunikációban, mind a minőség biztosításában súlyosan gátló tényező.

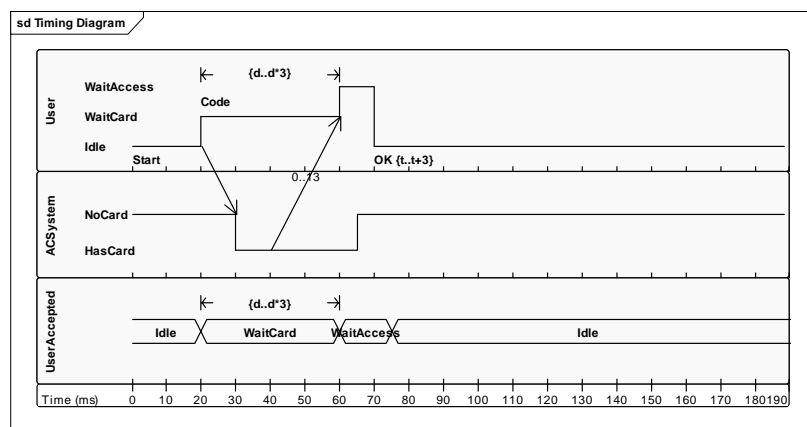
Az alábbi néhány példa ábra jól reprezentálja a szemléltetés jelentőségét:



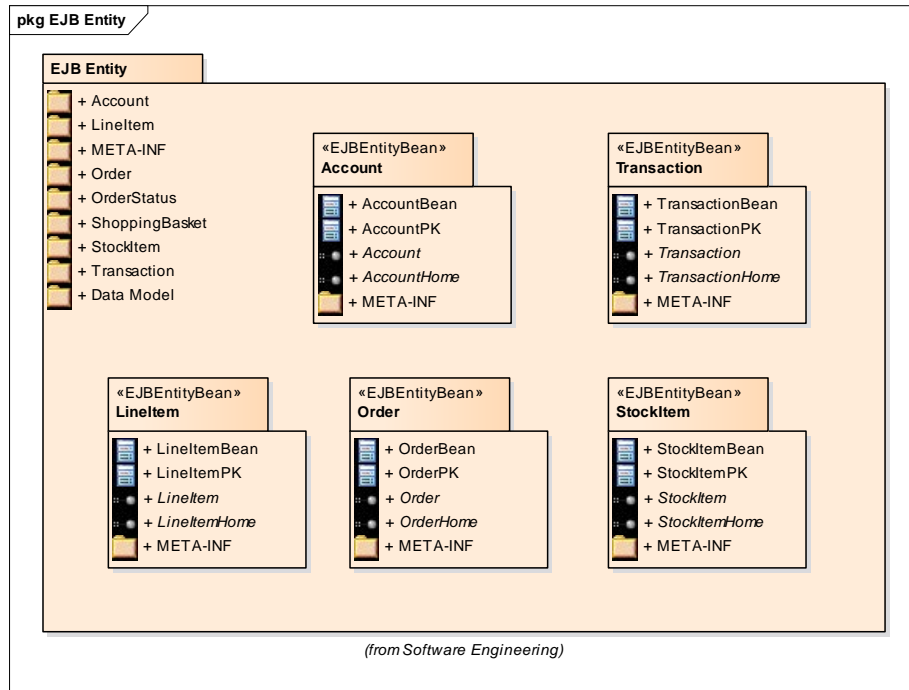
29. ábra: Szekvencia (sorozat-, „üzenetváltás”) modell



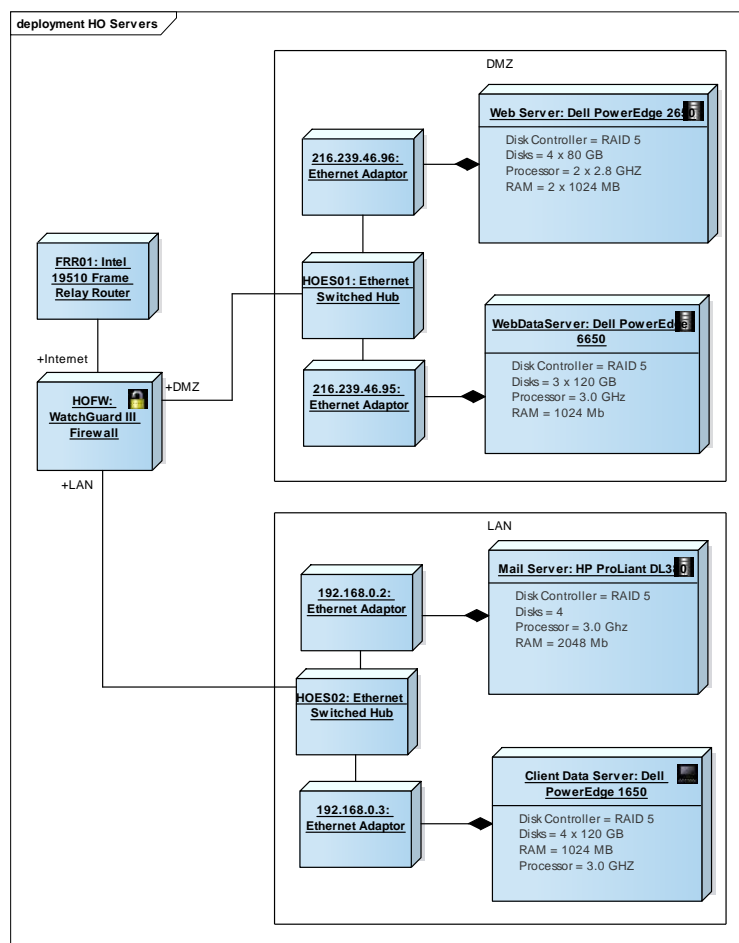
30. ábra: Tevékenység (Activity) modell



31. ábra: Időzítő (Timing) modell



32 ábra: Szoftver csomag modell



33. ábra: Telepítési modell

3.2.2. Fejlesztés

A fejlesztés minőség ellenőrzése során is hasonló elveknek kell érvényesülni, mint az elemzés, tervezés során. A tervezési megfontolásokat, a követelményeket szervesen (pl.: hivatkozások útján) össze kell kapcsolni a fejlesztési termékekkel, hogy követhető legyen a megvalósulási folyamat, és azonosítható legyen az egyes megoldási elemek létrejöttének szükségessége, valamint a teszteléshez beazonosítható legyen az elvárt funkcionalitásuk.

Architektúra minták, programkód sablonok, atomi programrész, és funkcionalitás generátorok alkalmazása alapvető követelményei a jó minőségnek.

A fejlesztési termékek moduláris felépítése a kezelhetőség, a továbbfejlesztés megkönnyítése miatt fontos jellemzők.

Formális követelménye minden fejlesztési munkának a programkódokban használt névalkotások szabályozása nevezéktan létrehozásával, illetve a közös megoldás elemek támogató modulokba gyűjtése, és a közös interfészek specifikálása, mindez túlnyomórészt még a fejlesztés megkezdése előtt. Ezek megléte szinte "magától" kikényszeríti az alapos, következetes, egységes megjelenést, illetve az "egyéni munkamódszerek" megfelelő korlátok közé szorítását.

A programkódok fejlesztő általi kommentálása szűrőpróbaszerűen ellenőrizhető, de ennél jobb megoldás a sablon alapú fejlesztés, aminek része a program modul magyarázó fejlécében, az alkalmazott speciális megoldások leírási lehetősége.

3.2.2.1. Fejlesztés támogatása

Csakúgy, mint a Tervezési eszközök, a modern fejlesztői környezetek (IDE) is szinte minden fejlesztési szintet, lépést támogatnak.

Fejlesztés támogató képességeik közül a legfontosabbak:

- Nemzetközi, és legjobb gyakorlat szabványok, minták alkalmazási lehetősége.
- Kódgenerálás, és visszafejtés.
- Adatszerkezet generálás és visszafejtés.
- Termék installációs csomag generálása.
- Kódváltozás követés.
- Csoportmunka támogatás.
- Egyéni munkakörnyezeti perspektíva alkalmazása.

3.2.3. Dokumentálás

A teljes anyag, amit itt közérdeklődésre bocsátottunk, tartalmaz egy dokumentum minta gyűjteményt, amelynek egyes darabjaiban, a lehetőségeink, és a jelen kereteink szerinti részletességre törekedve, azt a tartalomjegyzéket mutatjuk be, amely felölelni kívánja az egyes dokumentum típusok tartalmi elemeit.

Természetesen az egyes valódi dokumentumok nem tartalmazzák a mintákban előforduló összes lehetséges fejezetet, de lehetőséget adnak a velük való összevetésre, mind logikai, mind elrendezési szempontból. Minőség ellenőrzési szempontból az ellenőrizendő dokumentumok típusa, belső szerkesztettsége, elrendezése megfelelőnek kell, hogy mutasson a mintákkal.

3.3. Minőségi célok elérése

A tervezés az informatikai megvalósításoknál is alapvető szerepet játszik, és a jelentős anyagi befektetés miatt manapság erősen elvárt a minőség minél korábbi, a tervezési szakaszban már nagy pontossággal kimutatható megléte, várható szintje, vagyis az eredményesség garanciája. A tervek megfelelő szempontú vizsgálata, mint egyszerű eszköz, lehetővé teszi ennek a várható minőségnek a megvalósítást megelőző előrevetítését.

Az informatikai rendszereknek sokféle terve van, amik együttesen „Rendszerterv” összefoglaló néven közismertek. A rendszerek minőségi vizsgálatának kiinduló problémája, hogy milyen „tartalmi jegyekből”, és hogyan lehet következtetni, a tervben hosszasan leírt eredmény termék (valóban teljesülő) minőségére, lehetőleg még a megvalósulása előtt. Az alábbiakban egy ilyen ellenőrzési „Know How” vázlatát kívánjuk lefektetni.

A "Rendszerterv", akár fizikailag is elkülönülő dokumentumokat jelenthet, a különböző terv szintek, terv típusok mentén, de lényegét tekintve, logikailag egyetlen, összefüggő, egymásra épülő leírása a mindenkori, általában összetett számítástechnikai megoldás(ok)nak. Az összetettség természetesen nem csak ennek a területnek, iparágnak a jellemzője, de itt van a legnagyobb szabadsága a tervezőknek, és kivitelezőknek, mert a megvalósítási "közeg", azaz maga a szoftver, "tisztán" szellemi eszköz, és egyben termék, ennél fogva szinte végtelen lehetőséget ad, még a legapróbb részletek létrehozásában, kidolgozásában is.

A Rendszerterv azonban, bárhonnan nézzük is, egy műszaki leírás, és mint ilyen, hűen kell tükröznie az eredményt, a megvalósítandó / megvalósított rendszert. Mivel a szoftver termék is egyfajta gyártási folyamat útján születik, ezért igaz rá, hogy minél később derülnek ki a súlyos hibái, annál többre kerül a kijavításuk. Ezért olyan fontos,

hogy az előállítási folyamatában minél korábban észrevételre kerüljenek azok a jelenségek, amik ezekre a hibákra utalnak.

A tervekben megadott informatikai termék várható minőségének meghatározásához nem feltétlenül szükséges az informatika mélyreható ismerete. A terv minőségi megítélésének szempontjai ugyanis nem csak az informatikából erednek, hanem a természetes emberi gondolkodásból. A terv formális megközelítésével tartalmi megállapítások tehetők, a megállapítások láncolata pedig következtetéseket implikál. Ilyen módon egy, az informatikán kívüli szakma képviselője is érdemi nyilatkozatot tehet, de legalább is figyelemfelkeltő véleményt formálhat. Egy IPE szakember minőségi megállapításai, véleményezései, az informatikai munka támogatásában játszott szerepe legfontosabb feladatai közé tartozik.

A Rendszerterv fő fejezeteinek sorrendje logikailag egyezik a tervezési/megvalósítási folyamat feladatainak sorrendjével, így egymásra épülésük könnyen követhető.

Továbbiakban az informatikai rendszer minőségének Rendszerterv alapú vizsgálati szempontjait és módszereit foglaljuk össze.

(Az alábbi felsorolás, a teljesség igénye nélkül, az ellenőrzési szempontokat és módszereket, feladatokat gyakorlati okokból együtt tárgyalja.)

3.3.1. A tervek értelmezhetősége, a rendszer közérthetősége

3.3.1.1. Szempont

Elsődleges fontosságú, hogy a terv mind írásos, mind szemléltetett tartalma érthető, és egységesen értelmezhető legyen. A Rendszertervet minden felhasználójának értenie kell, méghozzá pontosan úgy, és pontosan azt, ahogy, és amit annak Szerzője gondolt. Ennek érdekében az írásos részeket a tervben közösen használt fogalmak magyarázatával (szótár), a szemléltetéseket pedig jelmagyarázatokkal kell ellátni.

A Rendszerterv egy dokumentum (gyűjtemény), terjedelme pedig megköveteli Tartalomjegyzékkel való támogatását. A jegyzéknek tükröznie kell a Szerző logikáját, amivel a tervet, és azon keresztül magának a rendszernek a készítési, bemutatási feladatait elgondolta, ugyanakkor szintén meg kell jelenítenie a mindenkori, széles szakmai közmegegyezésen alapuló, közkeletűnek tekintett, elfogadott szerkesztési elveket is.

A koncepcióalkotás, vagyis egy megoldás lényegének magyarázata, és ennek közvetítése mások felé, az informatikai rendszer végfelhasználója számára hatalmas súlyú tevékenység. A legkorszerűbb megoldású rendszer sem ér semmit, ha nincs, aki

használja. A koncepció bemutatása, és elfogadtatása ezért a sikeresség kulcsa, így hiánya előrevetíti a rendszer használatba vételének jelentős akadályát.

3.3.1.2. Módszertan, feladat

A tervnek tartalmaznia kell "Fogalom szótárt", vagy ennek megfelelő, a fent leírt fejezetet, önálló dokumentumot. A fogalmak osztályozásának létezése bizonyítja átgondolt használatukat, egy-egy véletlenszerűen választott definíció tartalmi, formai megjelenése az érthetőség, alaposság, pontosság elemeit tükrözi, vagy éppen ezek hiányát mutatja. A szótár fogalmainak a tervbeli előfordulásai szintén jól mutatják következetes használatukat. Egy ún. "nagy rendszer" a szótárában is terjedelmes, így ennek rövidege előrevetíti a pontatlanság, félreérthetőség lehetőségét, ami később a rendszer átvételi problémáihoz vezethet.

A "Koncepció" több alkalommal, különböző tervezési szinteken is előfordul. Szerepe azonban mindenütt ugyanaz: megjelenésében, tartalmában, érthetőségében összekötő kapocsként szolgál az egyik fél, a "megrendelő" elvárásai, és a megoldás készítő elképzelései között. Üzleti megrendelő esetén ezen a ponton "megy át" az Üzlet, az "Informatikába". A megrendelő Üzlet akkor lesz támogató, ha a Koncepcióban megtalálja az elvárásaira adott válaszokat, a "maga nyelvezetével" leírva. Ugyanez történik, a Fizikai szintre való átlépéskor, ahol a Tervező, mint "megrendelő", a Logikai elképzeléséből kialakított Fizikai terveit ismerteti a megoldás készítővel (pl.: programozó). A Koncepció ellenőrzése így szinte ugyanazt jelenti az IPE számára, mint az Üzlet számára: ha sikeres a megértés, ha "átmegy" az üzenet, akkor érthető, és (kisebb egyeztetésekkel) elfogadható, egyébként nem.

A közkeletű szerkesztési elvek alapján, a Tartalomjegyzékben önállóan elkülönítve, felismerhetőnek kell lennie:

- A Logikai, és a Fizikai terv leírások helyének (fejezet címben szerepelnek).
- A Szakterületi, vagy Működési/Üzleti modellnek, ami a "tisztá", az informatikai megoldással még NEM támogatott, alpműködési modellt vázolja.
- A "Szervezeti felépítés" (al)fejezetnek, ami az Üzleti oldal felelősségi körét írja le.
- A "Követelmény jegyzék" fejezetnek (önálló leírásnak), megbontva legalább "funkcionális" (mérhető), és "nem funkcionális" csoportosításban.
- A "Koncepció" alfejezetnek, a Logikai, Fizikai fejezetekben.
- A "Modell" alfejezetnek, a Logikai, Fizikai fejezetekben, ami szemléletesen mutatja be a rendszer elemeit, kapcsolatait.
- Az "Adatmodell" (al)fejezetnek, ami az adatok szerkezetét, kapcsolatait mutatja be.

- A "Támogatott/Üzleti folyamat" (al)fejezet(ek)nek, ami az informatikai megoldással támogatott üzleti (alap)folyamatot mutatja be.
- A "Működési/Üzleti szabály" (al)fejezet(ek)nek, ami a megfelelő működést biztosító, vagy az Üzlet által elvárt adat-, és működési szabályok pontos (lajstrom-szerű) jegyzéke.
- Az "Infrastruktúra" (al)fejezet(ek)nek.
- A "Tesztelés" (al)fejezet(ek)nek, amelyek (általában önálló dokumentum(ok)ként) a tesztelés/átvétel lépéseit írják le.
- Az "Üzemeltetés" (al)fejezet(ek)nek, ami az elkészült rendszer átadási körülményeit, napi működtetését írja le.

3.3.2. A tervezés követhetősége

3.3.2.1. Szempont

Tervezni egy eredmény elérésének minél egyszerűbb, és jobb (használhatóbb) módja miatt kell. A próbálkozások csökkentése előnyös, néha még életbevágó is. Nem csak az Informatikai munkamódszerre jellemző, hogy kitalálja a megoldást, és eltervezi az elérés lépéseit, ám biztosan csak az Informatika az a terület, ahol egy tervből számos megoldás születhet, mivel itt nincs a különböző megoldásokhoz szükséges, előre legyártandó, fizikai alkatrész, illetve könnyebb, egyszerűbb választani, és beszerezni az egyes "segédeszközöket".

A Követelmények meghatározása, majd a Logikai, és az erre épülő Fizikai tervezés, hasonlóan más mérnöki területek gyakorlatához, itt is különválnak, és a Konceptió fogalmával írható le az egyikből a másikba történő átmenet szabályozottsága, annak elfogadásával, hogy ellenkező irányba indulva már nincs ilyen egyértelmű kapcsolódás (pl.: Entitás, al-Entitás logikai szintű modellje legalább 3 féle fizikai megvalósításra képezhető le). A lényeg az egyes tervezési szintek közötti átmenet szabályainak rögzítésében van.

Nem csak egy rendszer megértéséhez, hanem továbbfejlesztéséhez is szükséges a rendszer "milyenségének" ismerete. Amennyiben ugyanis ismertek az átmenet (leképezés) szabályai, akkor azokon a pontokon új, vagy módosított szabályokkal más elrendezés alakítható ki.

3.3.2.2. Módszertan, feladat

A Rendszertervnek (akár, mint szimbolikus leírás gyűjteménynek) tartalmaznia kell a Követelményeket, a Logikai, és az ebből származtatott Fizikai terveket, valamint ezek akár tételes, akár koncepcionális kapcsolatát.

3.3.3. A tervezési elvek következetessége

3.3.3.1. Szempont

A tervezési szintek egymásból következnek, ezért szerves a kapcsolatuk. A Követelmények (funkcionális, azaz "mérhető" típusai) határozzák meg az egyes szinteken létrehozott megoldásokat, így a függési lánc tetején állnak. A függés, vagy más szóval kapcsolat megjelenésének hiánya felveti az Üzleti igények meg nem értését, vagy részlegességét, ez pedig a megvalósítás hiányosságát, vagy rossz esetben irreleváns (használatatlan) voltát. Ez a pont az, ahol egy Rendszer tervezésének az "ütőere" van. A következetesség ugyanis átgondoltságot, és egyben mások általi követhetőséget biztosít, ami mindkét félnek (megrendelő, és kivitelező) előnyére szolgál.

3.3.3.2. Módszertan, feladat

A tervezési szintek egymásból következésének, szerves kapcsolatának ún. (vissza)hivatkozások formájában meg kell jelennie a szintek tárgyalásában. A legtöbbet hivatkozott, és minden szinten biztosan megjelenő hivatkozás a Követelmények szintjére történik. A legtöbb esetben tételes a hivatkozás.

(pl.: Teszteset → Követelmény), de pl. a Logikai adatmodell Entitás koncepciója megegyezéses, vagy "szimbolikus" alapú. Abszolút tételes egyezés szükséges azonban az Üzleti szabályok megvalósítási tervei esetében. Ezek megléte különösen a tesztelésükkor kap nagy hangsúlyt.

3.3.4. Korszerűség, megbízhatóság, nívó

3.3.4.1. Szempont

A szoftverfejlesztés minősége hatékonyabban garantálható, az iparágban, a szakmai közösségek együttműködésével folyamatosan kialakuló, tisztuló, és jól tesztelt megoldási sablonok, az ún. "tervezési minták" felhasználásával. A mintának az (eredeti) architektúrális specifikációja szerinti, megfelelő (pontos) alkalmazása egyaránt komoly biztosíték használatának szükségességére, valamint a kívánt cél megbízható elérésére is. Összetett, többszörösen rétegzett, korszerű fejlesztő nyelv, és keretrendszer segítségével megvalósított termékek esetén alkalmazása szinte alapvető architektúrális követelmény, ezért elmaradása egyfelől felveti a megoldás korszerűségének, és megbízhatóságának a hiányát, másfelől pedig a fejlesztési költségek esetleges szükségtelen növekedését.

3.3.4.2. Módszertan, feladat

Amennyiben nincs használatban közismert (szakmán belül jegyzett, jó megítéléssel rendelkező, széles körben bizonyítottan jelentős) ún. Megoldási keretrendszer (Solution Framework), ami akár a megoldás szállító saját fejlesztése is lehet, vagy nincs használatban Megoldási sablon gyűjtemény az egyes kritikus megoldási elemek létrehozásában, vagy a "sablonizálás" teljes hiánya áll fenn, akkor a nívó szintje nem biztosított. Természetesen létezhet teljesen egyedi megoldás is, de ezt a munka kezdetén hangsúllyal kell deklarálni, és a kockázatait megbecsülni. Minden más esetben a szakszerűséget kell erősen vizsgálni.

ÁBRAJEGYZÉK

1. ábra: UML diagramm paletta
2. ábra: A Digitális Világ, az ember által teremtett Virtuális Valóság
3. ábra: Zachman kategorizációs rendszere, informatikai fejlesztésre alkalmazva
4. ábra: Az ajánlott specifikációs sablonok informatikai besorolási rendszere
5. ábra: Funkcionális követelmény modell
6. ábra: Nem Funkcionális követelmény modell
7. ábra: Követelményminősítés modell
8. ábra: Követelmények színkódolása
9. ábra: Követelmények származási modell
10. ábra: Esettanulmány modell (Use Case)
11. ábra: Esettanulmány részletező
12. ábra: Esettanulmány - összetett
13. ábra: Állapotgép
14. ábra: Projekt felelősségi modell
15. ábra: Üzleti folyamat modell – egyszerű
16. ábra: Üzleti folyamat modell – összetett
17. ábra: Megvalósítandó Üzleti (cél)folyamat modell
18. ábra: [Fogalmi-, Domain-,] Szakterületi modellek (konceptcionális)
19. ábra: Fogalmi (konceptcionális) adatmodell
20. ábra: Felhasználói felület (User Interface - UI) Logikai modell
21. ábra: Logikai összetevő (Component) modellek
22. ábra: Logikai adatmodell
23. ábra: Hálózati Fizikai modell
24. ábra: Fizikai adatmodell
25. ábra: Teszt terv
26. ábra: A SparX Systems “Enterprise Architect” tervező eszköze
27. ábra: A Sybase “Power Designer” tervező eszköze
28. ábra: Folyamatelemző (SparX : Enterprise Architect)
29. ábra: Szekvencia (sorozat-, „üzenetváltás”) modell
30. ábra: Tevékenység (Activity) modell

31. ábra: Időzítő (Timing) modell

32. ábra: Szoftver csomag modell

33. ábra: Telepítési modell

A sorozat keretében eddig megjelent kiadványok

1. Dr. Barna Lajos, Eördöghné Dr. Miklós Mária, Dr. Szánthó Zoltán, Dr. Balla József, A biztonságos ivóvízellátás megteremtésének tervezési eszközei
2. Dr. Szilágyi Zsombor, Dr. Szunyog István Mérések a gáziparban
3. Németh András, Milávecz Richárd Iparban használatos vízminőségek
4. Borbás Lajos Dr. Felépítéselvű (additív) gyártástechnológiák a gépészetben
5. Berencsi Miklós, Bereczky Ákos, Horváth László, Kovács Gergely, Mihálffy Krisztina Kerékpárosbarát közlekedéstervezés