



Rendszerfejlesztés és tesztelés

Dr. Csertán György

Frits Márton

Ismertető

(tárgykövetelmények)

Bemutakozás

Dr. Csertán György

- csertan.gyorgy@dcs.uni-pannon.hu
- I. ép. 918
- csütörtök, péntek

COVID-19 miatt a személyes találkozást lehetőleg kerüljük

Cél

Előadás:

- Az alapvető rendszerfejlesztési és tesztelési módszerek bemutatása.

Gyakorlat:

- A tanultak alkalmazása egy gyakorlati projekten.

Gyakorlati jegy

- ZH dolgozat – 50%
- projekt feladat – 50%

Előadás

Rendszerfejlesztés alapjai

Verifikáció és validáció

Szoftvertesztelés

Specifikáció alapú tesztelés

Kód alapú tesztelés

Gyakorlat („Szoftvertchnológia” tárgy)

UML

- Használati eset diagram
- Tevékenység diagram
- Osztály diagram
- Szekvencia diagram

Projekt feladat (3 fős csoportok)

- Specifikáció
- 4 db dokumentáció (1-1 UML diagram típusról)
- 1 db működő szoftver

Követelmények

Óra látogatás

- Előadás – ajánlott
- Gyakorlat – beszámolókon megjelenés kötelező

Beszámolók értékelése

- nem felelt meg (1) / megfelelt (4) / kiválóan megfelelt (5)

Zárthelyi pótlása

- vizsgaidőszak első 12 napján belül

Zárthelyi eredményének kialakítása

- 0-49% - elégtelen / 50-65% - elégséges / 66-80% - közepes / 81-90% - jó / 91-100% jeles

Követelmények (folyt.)

Aláírás feltétele

- legalább megfelelt beszámolók
- félévközi ZH legalább 40%-os

Megtagadott aláírása pótlása

- vizsgaidőszak első 12 napján belül előre megadott időpontban

Ajánlott irodalom

Ian Sommerville: Szoftverrendszerek fejlesztése. Panem kiadó, 2007. ISBN: 9789635454785

- az előadás rendszerfejlesztés része a 2. fejezetre épül
- az előadás tesztelés elméleti része a 22. és 23. fejezetekre épül

UML@Classroom slides

- a gyakorlaton az UML részhez
- <http://www.uml.ac.at/en/lernen>

Beszédes Árpád, Gergely Tamás: Tesztelési módszerek. Typotex kiadó, 2011.

- az előadás tesztelés a gyakorlatban részéhez

Rendszerfejlesztés alapjai

(elméleti áttekintés)

Mit értünk rendszer alatt?



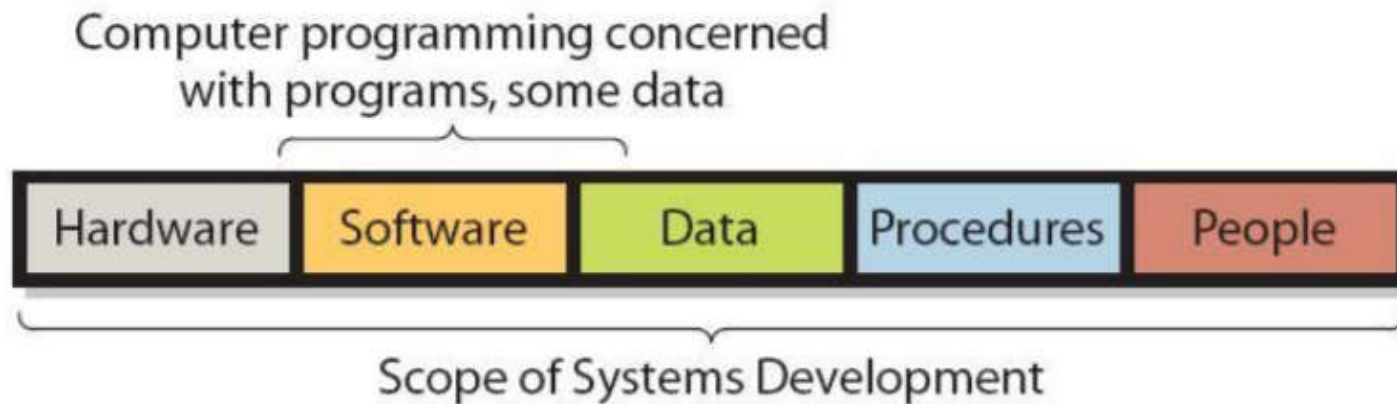
Kapcsolódó komponensek halmaza, amelyek egy közös cél érdekében működnek együtt.

Az egyes rendszerkomponensek más rendszerkomponensektől függnnek.

A rendszerkomponensek tulajdonságai és viselkedése elválaszthatatlanul összefonódnak.

Mi a rendszerfejlesztés?

Hardver, szoftver, adatok, eljárások, felhasználók létrehozása / összehangolása.



Rendszerkategóriák

Technikai (számítógép-alapú) rendszerek

- Hardvert és szoftvert tartalmazó rendszerek. Az operátorokat és a rendszert működtető eljárásokat nem tekintjük a rendszer részének.

Ember-gép rendszerek

- Olyan rendszerek, amelyek technikai rendszereket is tartalmaznak, csakúgy, mint a rendszert működtető eljárásokat és a technikai rendszerrel kapcsolatot tartó embereket. Az ember-gép rendszereket különféle szervezeti szabályok és irányelvek szabályozzák.

Az ember-gép rendszerek jellemzői

Globális (eredő) tulajdonságok

- Olyan rendszertulajdonságok, amelyek a rendszerkomponensektől és azok kapcsolatától függenek.

Nem determinisztikus viselkedés

- Azonos bemenőjelre nem mindig azonos kimenőjelet produkálnak, mert a rendszer viselkedése részben az emberi operátoroktól függ.

Szervezeti céloktól való komplex függés

- Nem csak a rendszertől függ, hogy az mennyire képes a szervezet céljait szolgálni.

Eredő tulajdonságok

Az egész rendszerre vonatkozó tulajdonságok, melyek nem származtathatók a komponensek tulajdonságaiból

A globális (eredő) tulajdonságok a rendszerkomponensek kapcsolatából adódnak

Tehát ezen jellemzők csak akkor mérhetők, ha a komponensek rendszerré történő integrációja megtörtént

Példák eredő tulajdonságokra

Tulajdonság	Leírás
Térfogat	A rendszer teljes térfogata a komponensek összeállításának mikéntjétől függ
Megbízhatóság	A rendszer megbízhatósága függ a komponensek megbízhatóságától, de váratlan egymásrahatások újabb típusú hibákat okozhatnak.
Biztonság	A rendszer biztonsága (támadások elleni ellenálló képessége) komplex, nehezen mérhető tulajdonság. Újabb típusú támadásokra a rendszertervezők nem számíthattak, így a beépített biztonsági mechanizmusokat ezek ki tudják játszani.
Javíthatóság	Jellemzi, hogy milyen egyszerű a rendszert javítani, miután a hibát észlelték. Függ a rendszer diagnosztizálhatóságától, a hibás komponensek elérhetőségétől, módosíthatóságától és cserélhetőségétől.
Használhatóság	Milyen könnyű a rendszert használni. Függ a technikai rendszer komponenseitől, az operátoroktól, valamint a működtető környezettől.

Az eredő tulajdonságok típusai

Funkcionális tulajdonságok

- Akkor láthatók, ha a rendszer valamennyi eleme egy cél elérése érdekében közösen dolgozik. Példa: Egy kerékpárnak akkor funkcionális tulajdonsága, hogy közlekedési eszköz, ha azt alkatrészeiből már összeszerelték.

Nem-funkcionális tulajdonságok

- Pl.: megbízhatóság, teljesítmény, biztonság. Ezek a rendszernek a környezetével való kapcsolatát jellemzik. Számítógépes rendszereknél gyakran kritikus tulajdonságok: amennyiben egy minimális szintet nem érik el, a rendszer könnyen instabillá válhat.



Rendszerfejlesztés – precízebben

Ember-gép vagy technikai rendszerek specifikációja, tervezése, implementációja, validációja, telepítése és fenntartása.

Foglalkozik a rendszer által nyújtott szolgáltatásokkal, a létrehozást és működést befolyásoló kényszerekkel, valamint a felhasználás módjával.

A rendszerfejlesztés folyamata

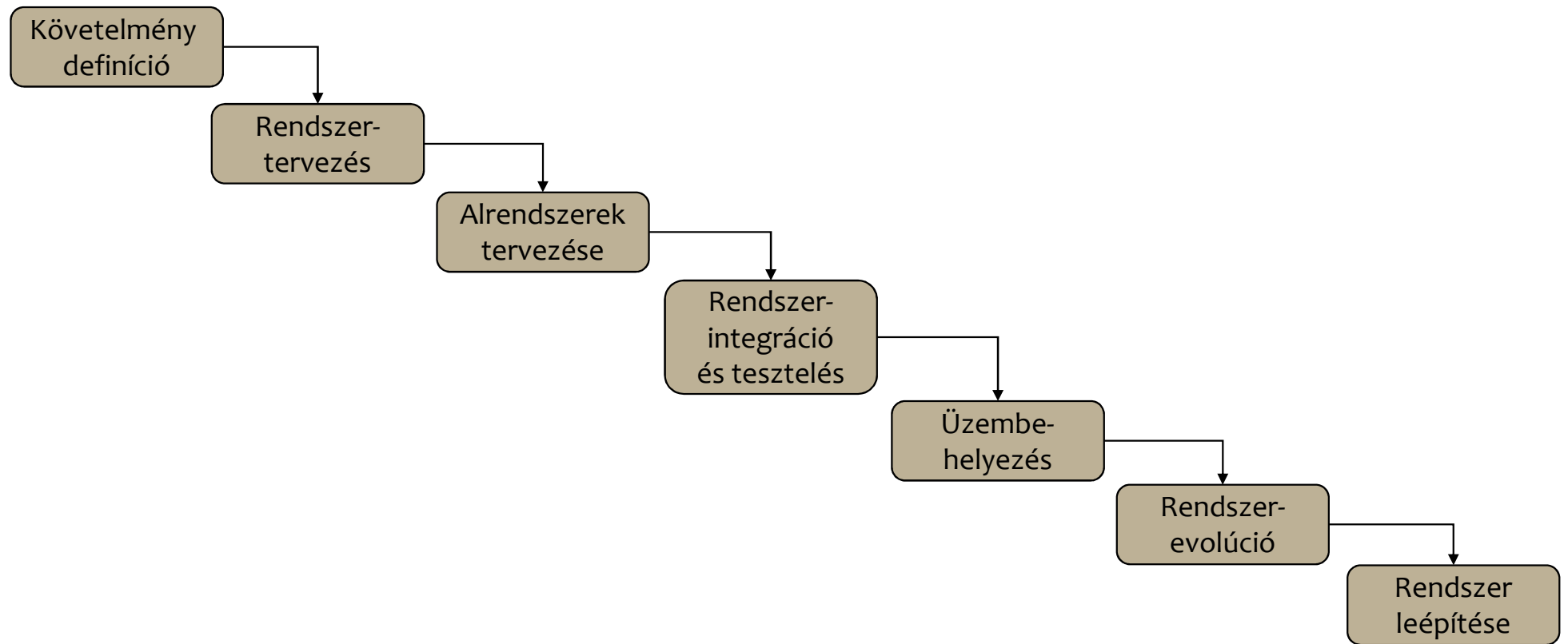
Általában valamilyen modellt követ

- az egyes tevékenységek sorrendjét definiálja

Vízesés modell (legalapvetőbb modell, de sok másik is létezik)

- rendszer különböző részeinek párhuzamos fejlesztése
- fázisok között csak kis iterációs lehetőségek
 - hardver változtatása drága
- különböző szakterületek együttműködése
 - sok lehetőség a félreértésekre

A rendszerfejlesztés folyamata



Rendszerkövetelmények

A követelmények három típusa

- Absztrakt funkcionális követelmények. A rendszer funkcióit absztrakt módon definiáljuk
- Rendszertulajdonságok. Az egész rendszerre vonatkozó nem funkcionális követelményeket definiáljuk.
- Nem kívánatos tulajdonságok. Nem megengedett viselkedés specifikációja.

Definiálni kell a rendszer helyét és célját is a felhasználó szervezeti egységben.

A rendszer célja

Definiálni kell, hogy miért van szükség a rendszerre az adott környezetben.

Funkcionális célok

- Pl.: „Tűzvédelmi és behatolás jelző rendszer, amely külső és belső riasztási jeleket ad tűz, vagy illetéktelen behatolás esetén”

Szervezeti célok

- Pl.: „Biztosítani kell, hogy a normál munkavégzés folyamatát komolyabb mértékben nem zavarják meg olyan rendkívüli események, mint tűzeset, vagy illetéktelen behatolás”

Problémák a rendszerkövetelmények körül

Komplex rendszerek általában nehéz problémák megoldását tűzik ki célul

- A probléma nem teljesen ismert;
- Specifikáció közben változik a probléma.

A rendszer életciklusa alatt számítani kell a hardver és a kommunikációs rendszer fejlődésére.

A nem funkcionális követelmények definiálása nehéz, különösen, ha a rendszer felépítése, komponensei nem ismertek.

A rendszertervezés folyamata

A követelmények csoportosítása

- Követelményeknek kapcsolódó csoportokra osztása

Alrendszerek meghatározása

- Alrendszerek olyan halmazának meghatározása, amelyek együttesen képesek a rendszerkövetelmények teljesítésére.

Követelmények hozzárendelése az alrendszerekhez

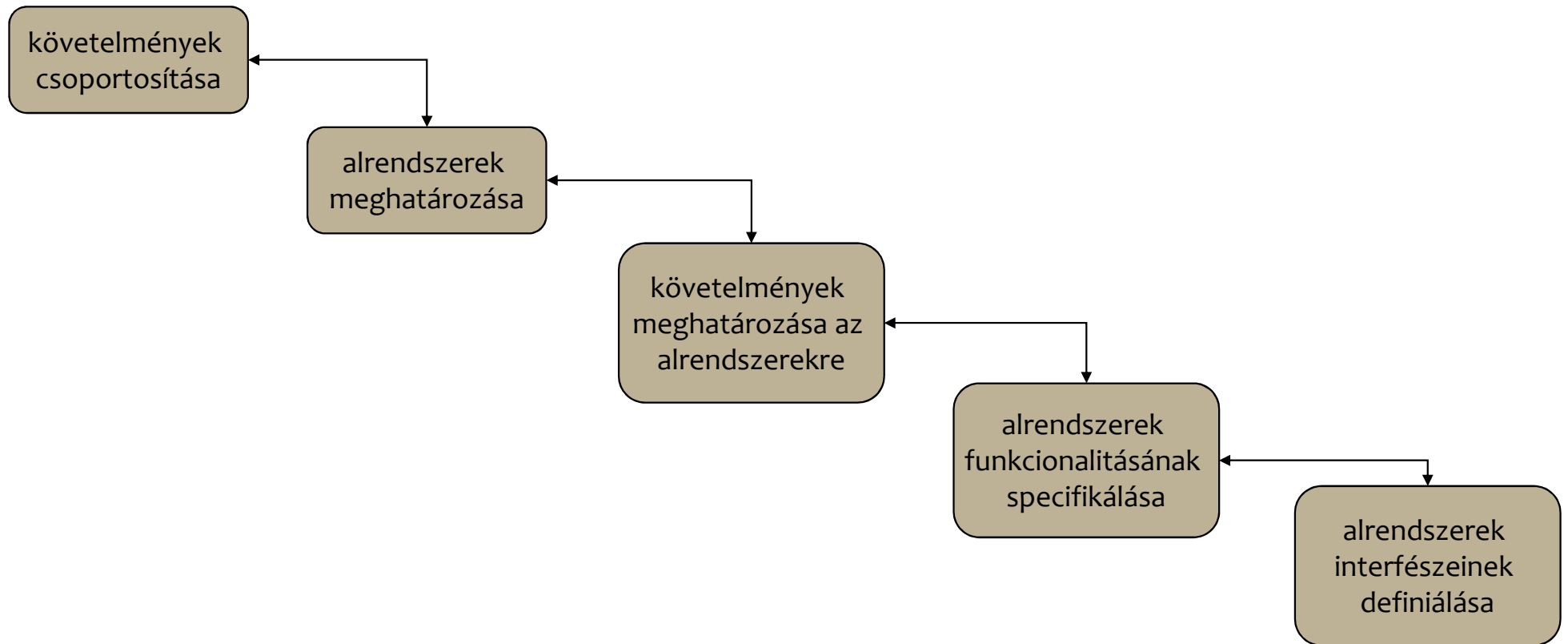
- Nehézségbe ütközik, ha COTS rendszereket integrálunk.

Alrendszerek funkcionalitásának specifikálása.

Alrendszerek interfészeinek definiálása

- Különösen fontos párhuzamos alrendszer-fejlesztés esetén.

A rendszertervezés folyamata



A rendszertervezés nehézségei

Hosszas viták előzhetik meg a hardver-szoftver-emberi erőforrásokra való dekompozíciót.

Gyakori tévhit, hogy a nehéz tervezési problémákat szoftverben könnyű lesz megoldani.

A hardver platformok gyakran nem elégítik ki a követelményeket, amit aztán a szoftvernek kell kompenzálnia.

Követelmény- és rendszertervezés

A követelménytervezés és a rendszertervezés szorosan összefügg.

A környezet és más rendszerek szűkítik a tervezési lehetőségeket. Lehet, hogy követelmény egy adott rendszer felhasználása.

Egy kezdeti rendszerterv szükséges lehet a követelmények rendszerezéséhez.

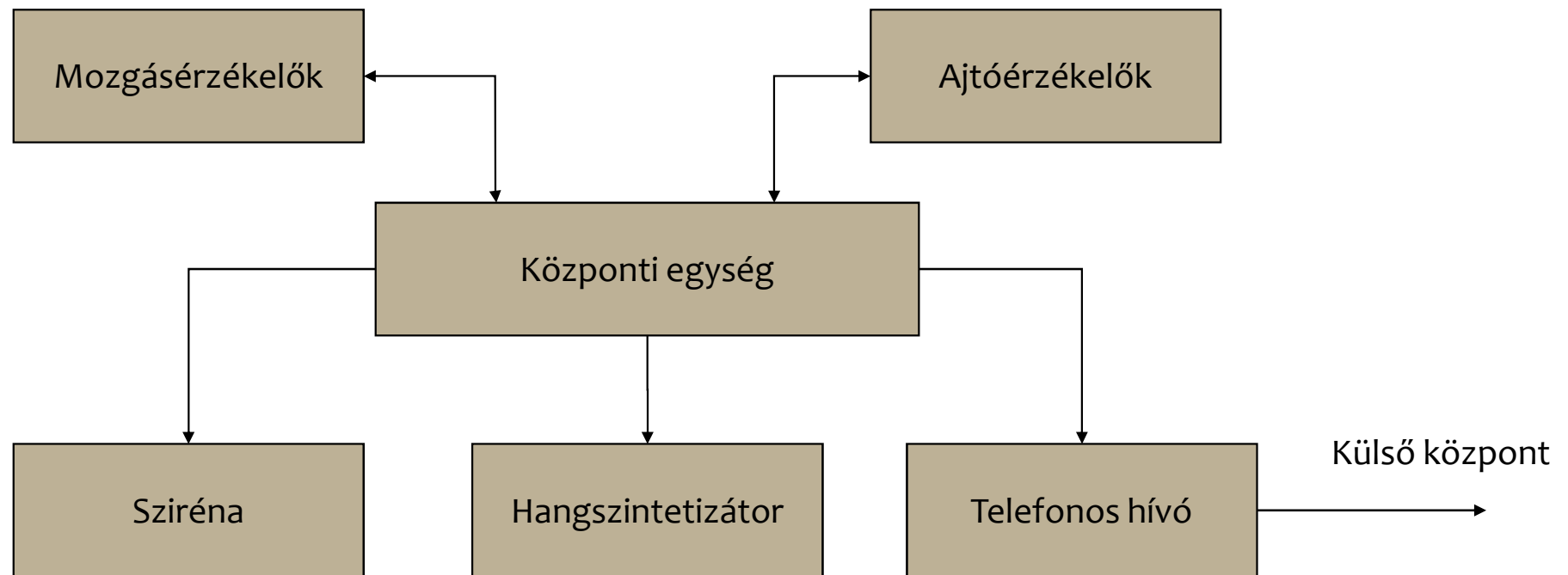
A rendszer tervezése közben egyre több információ gyűlik fel a követelményekről.

Rendszermodellezés

Az architektúra modell a rendszert alkotó alrendszerek absztrakt reprezentációja

Tartalmazhatja az alrendszerek közötti főbb információfolyamokat is
Általában blokk-diagram formájában használjuk

Példa: betörésjelző rendszer



Alrendszerek leírása

Alrendszer	Leírás
Mozgásérzékelők	A monitorozott helységekben mozgást érzékel
Ajtószenzorok	Érzékeli az épület külső ajtóinak nyílását
Központi egység	A rendszer működését vezérli
Sziréna	Hangjelzést ad behatolás esetén
Hangszintetizátor	Hangüzenetet szintetizál a behatoló feltételezett helyéről
Telefonos hívó	Hívásokat generál pl. a rendőrség, biztonságiak, stb. felé

Alrendszerek fejlesztése

Általában párhuzamosan zajlik a hardver, szoftver és a kommunikáció fejlesztése.

Tartalmazhat COTS (Commercial Off-The-Shelf) rendszerek beszerzését is.

A fejlesztő csoportok között nincs kommunikáció.

Amennyiben változtatásra van szükség, a lassú és bürokratikus engedélyeztetési eljárások miatt gyakran határidő módosítás is szükséges.

Rendszerintegráció

Az a folyamat, amelynek során a hardver, szoftver és személyi állomány együttesen rendszert alkot.

Célszerű inkrementálisan végezni (egyszerre csak egy alegység integrálása).

Az alegységek közötti interfész problémák rendszerint ebben a fázisban derülnek ki.

A rendszerkomponensek koordinálatlan beszállítása gondokat okoz.

Tesztelés

Az a folyamat, amelynek során megvizsgáljuk, hogy a rendszerben van-e hiba.

A lehető legalaposabb vizsgálatot kell elvégezni, amit az idő és a költség enged.

A tesztelés szinte külön szakma, nagy rendszereknél külön csapat végzi.

Sokrétű tevékenység, később látjuk majd.