



IT ADVISORY

# Bombriadó nélkül

biztonságos rendszerek fejlesztése –  
a rendszerfejlesztés biztonsága

Gaidosch Tamás  
partner

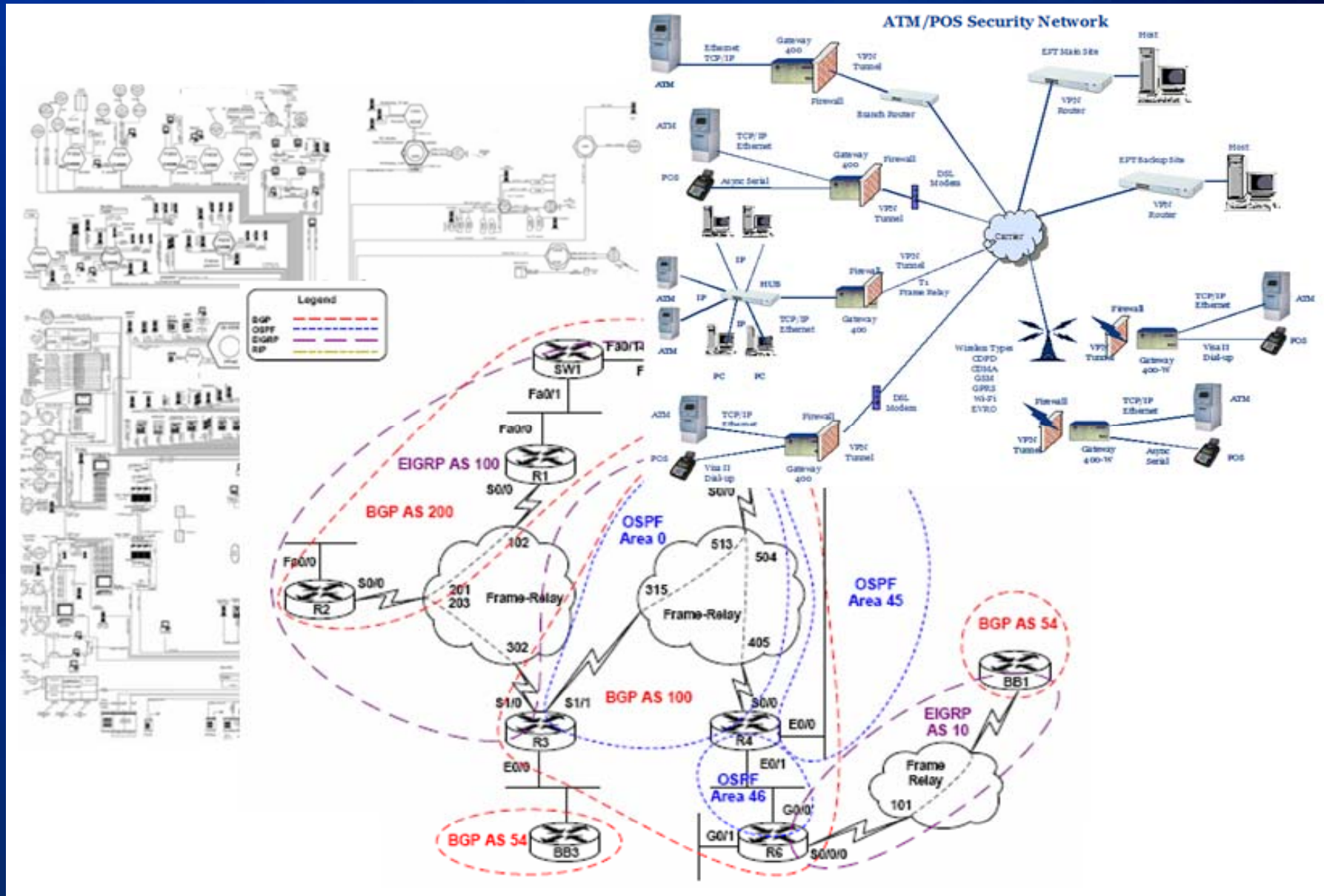
2010.05.27.

# Tartalom

---

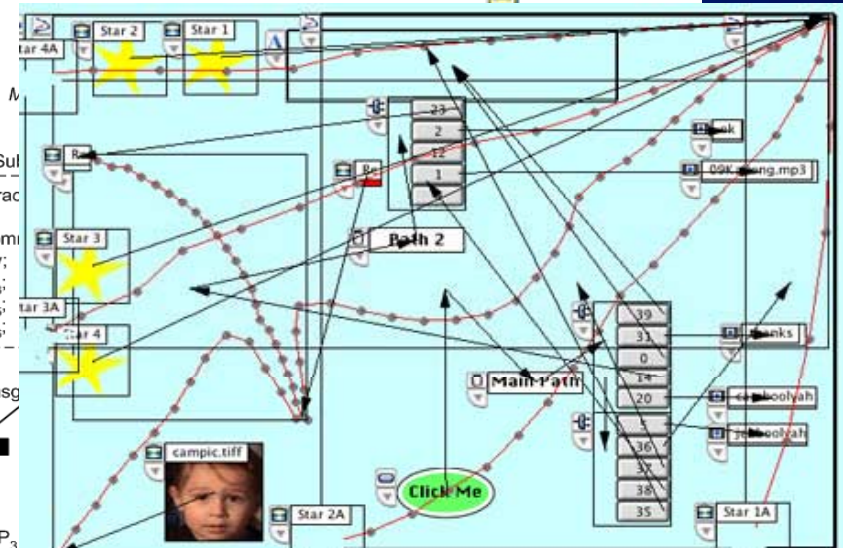
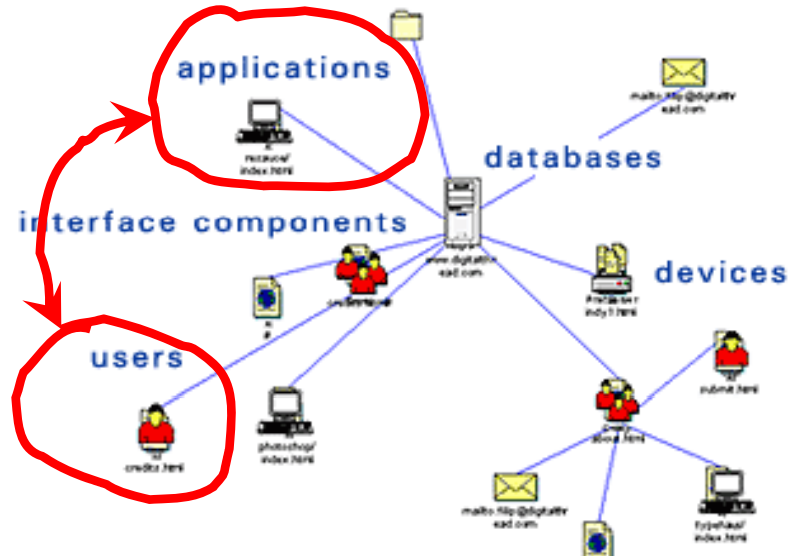
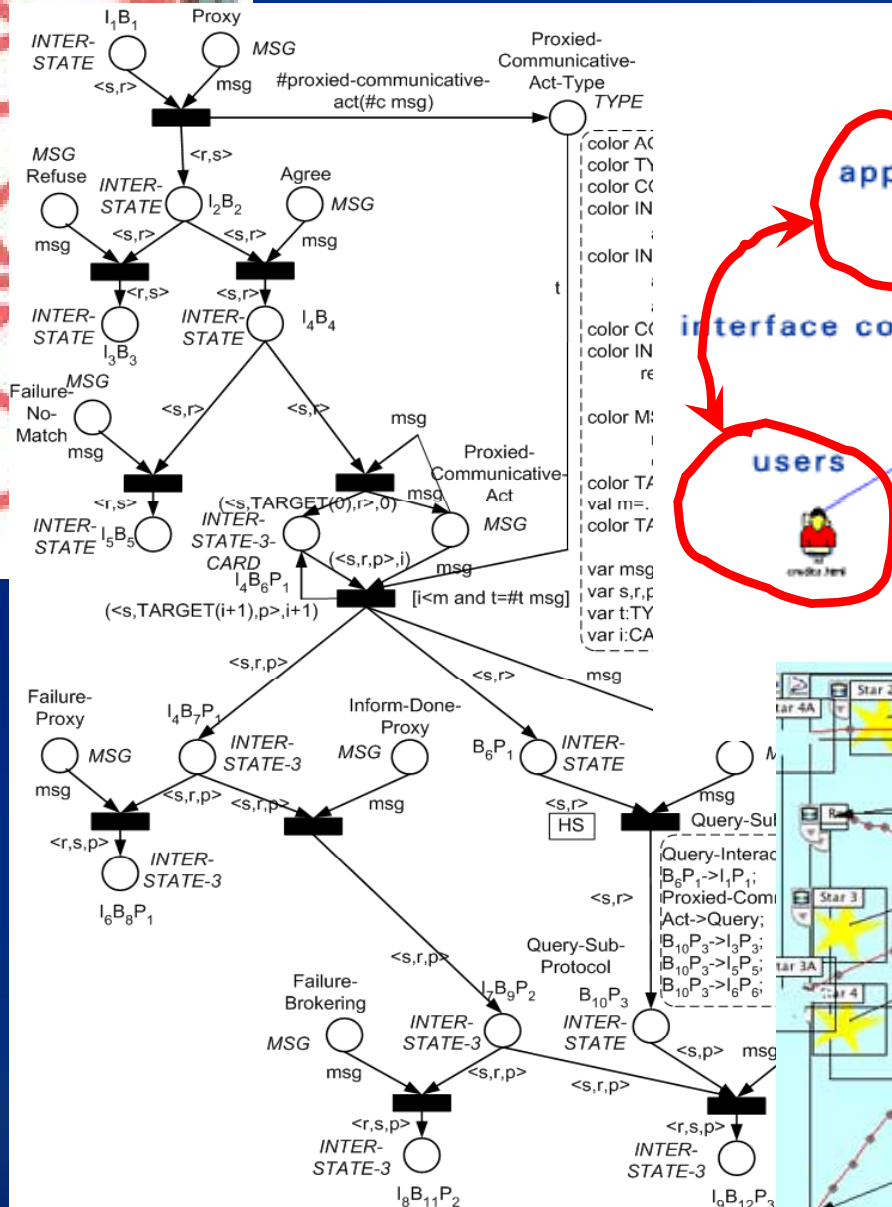
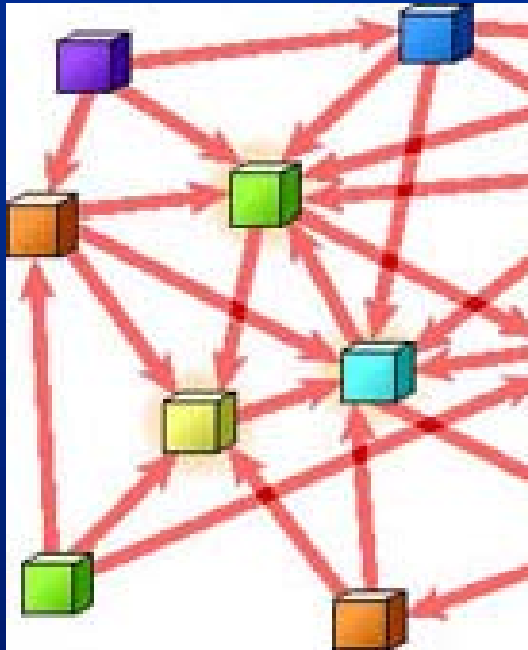
- ◆ Biztonság és SDLC
- ◆ Analógiák a biztonsági tervezéssel
- ◆ Esettanulmány
- ◆ Tanulságok

# Rendszer komplexitás

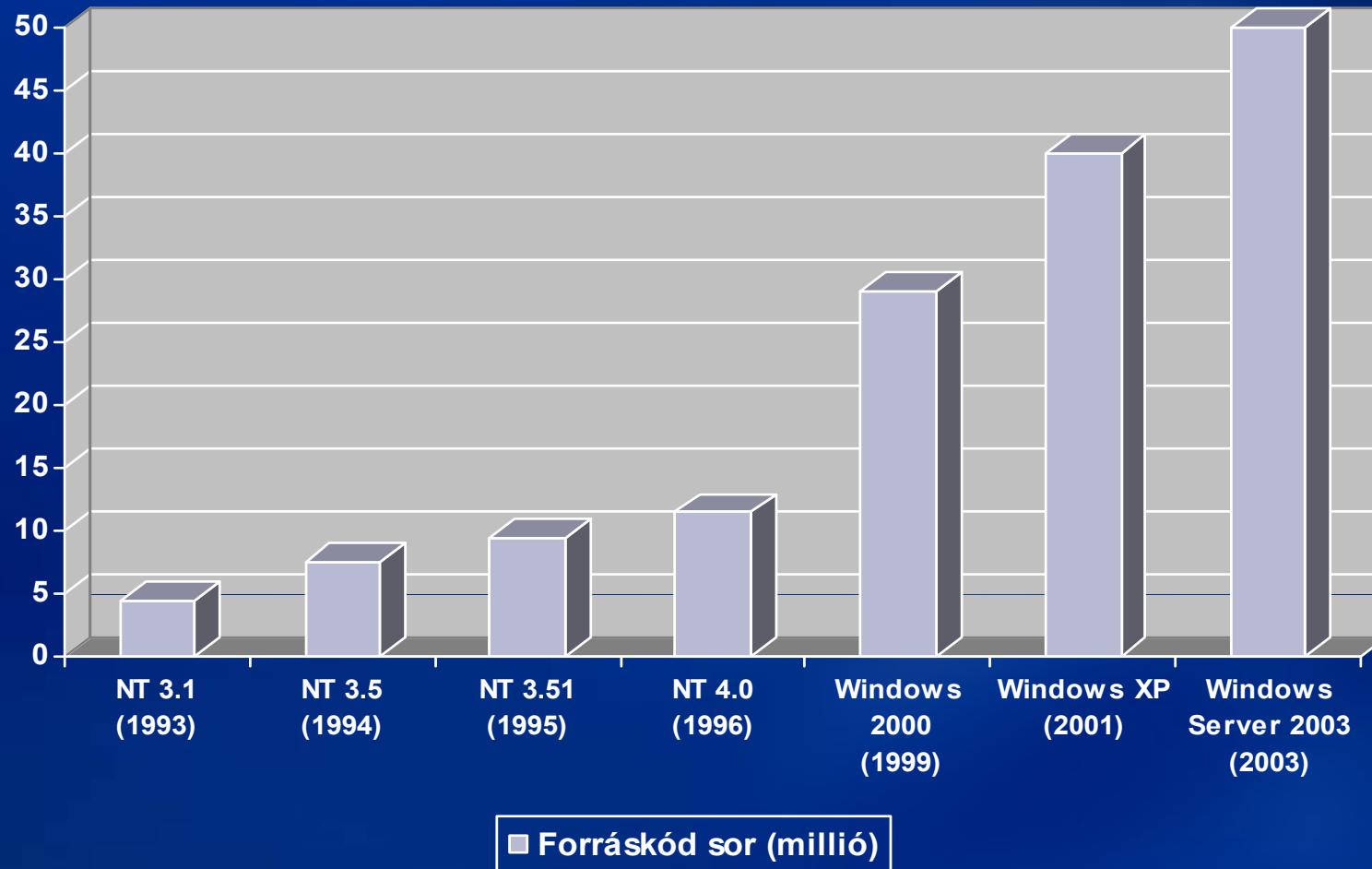




# Interakció komplexitás

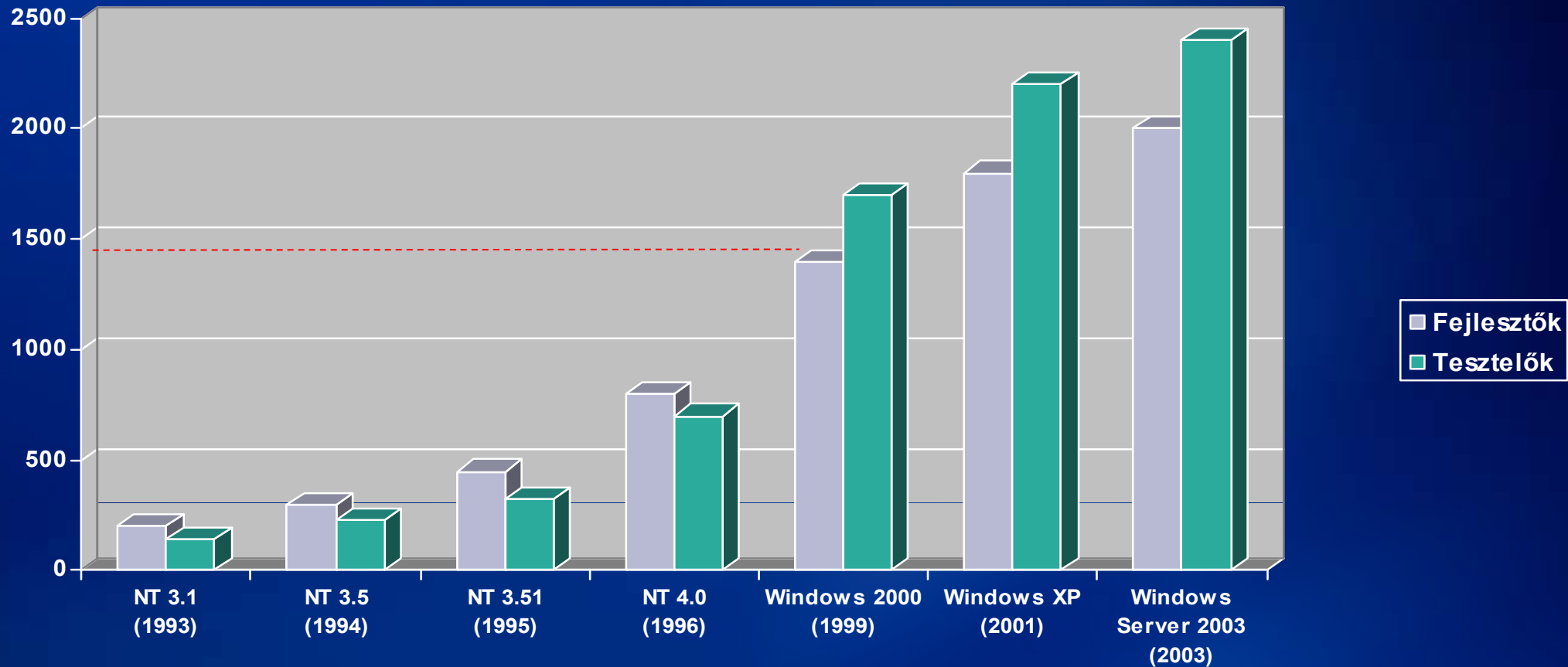


# Szoftver komplexitás



Vincent Maraia: The Build Master

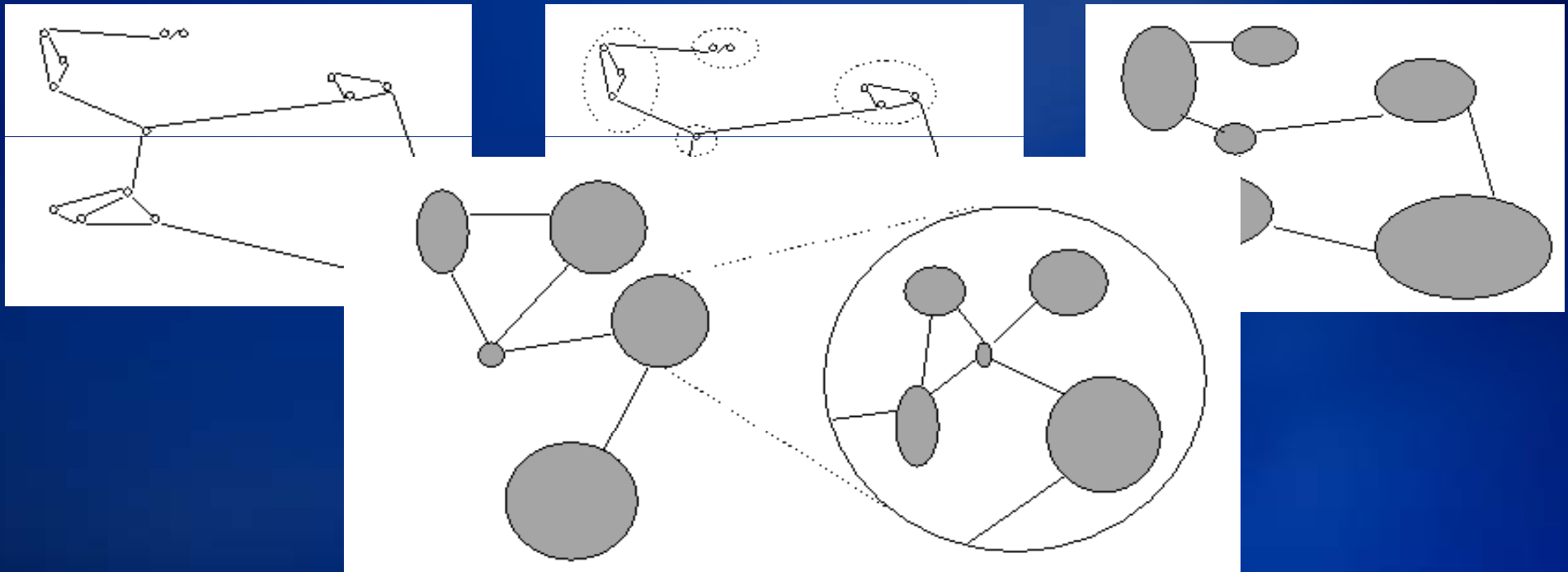
# Szoftver komplexitás



Vincent Maraia: The Build Master

# Hogyan kezeljük a komplexitást?

- **Miért van szükség módszeres megközelítésre?**
  - közvetlen memóriánk csak kb. 7 objektumot tud kezelni
- **Csak két általános módszer létezik**
  - oszd meg és uralkodj
  - absztrakció



# Biztonságos rendszerek fejlesztése - kihívások

- **A komplexitás a biztonság ellen hat**
  - Biztonsági tervezési alapelv: **keep it simple!**
- **A komplexitás a karbantarthatóság ellen hat**
- **A biztonság három aspektusát és egymásra hatásukat kell tudnunk kezelni**
  - Technológia és rendszerek
  - Üzleti folyamatok
  - Emberek
- **Mindhárom aspektusban lépést kell tartani a folyamatos változással**

**KISS**

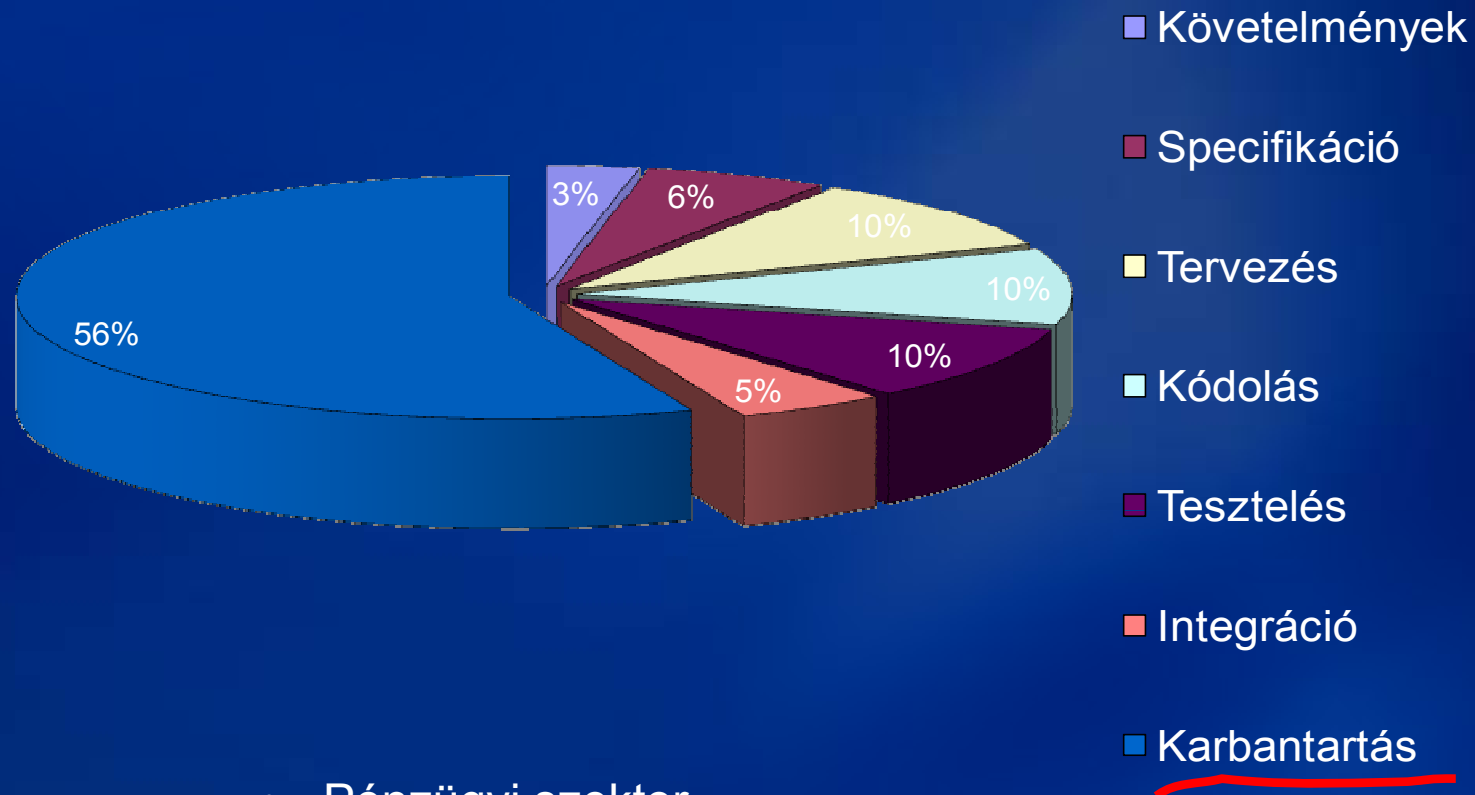
# Rendszerfejlesztési életciklus modellek (SDLC)

- A rendszerfejlesztési projekt tevékenységeinek definíciója és egymásutánisága
- „A rendszerrel kapcsolatos tevékenységek összessége, ami magába foglalja a rendszer koncepciótervezését, fejlesztését, implementációját, üzemeltetését, karbantartását, és végül használaton kívül helyezését, mely egy újabb rendszer koncepciótervezését indítja el” (NIST SP 800-34)
- Általános keret, projektenként kell testre szabni

# Mit akarunk elérni egy SDLC-vel?

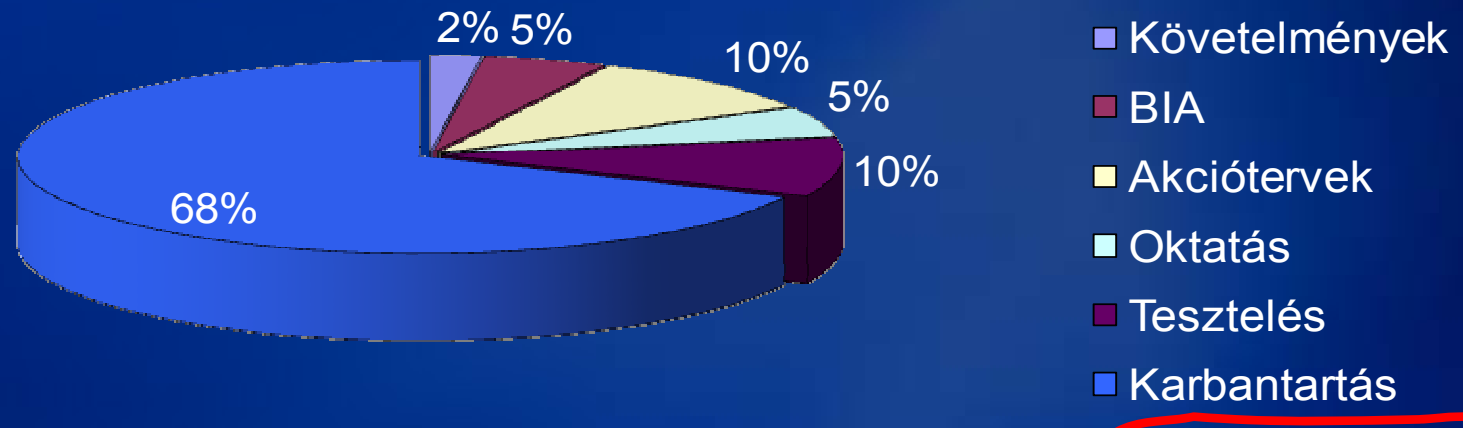
- **Olyan rendszert kifejleszteni, amely**
  - megfelel a követelményeknek, beleértve az idő- és költségkereteket
  - karbantartható

# Költségek egy tranzakciós rendszer életciklusában (TCO %)



- Pénzügyi szektor
- Hat éves periódusra számolva
- Átlagosan változó környezet

# Költségek egy BCP életciklusában (TCO%)



- Pénzügyi szektor
- Hat éves periódusra számolva
- Átlagon felüli ütemben változó környezet

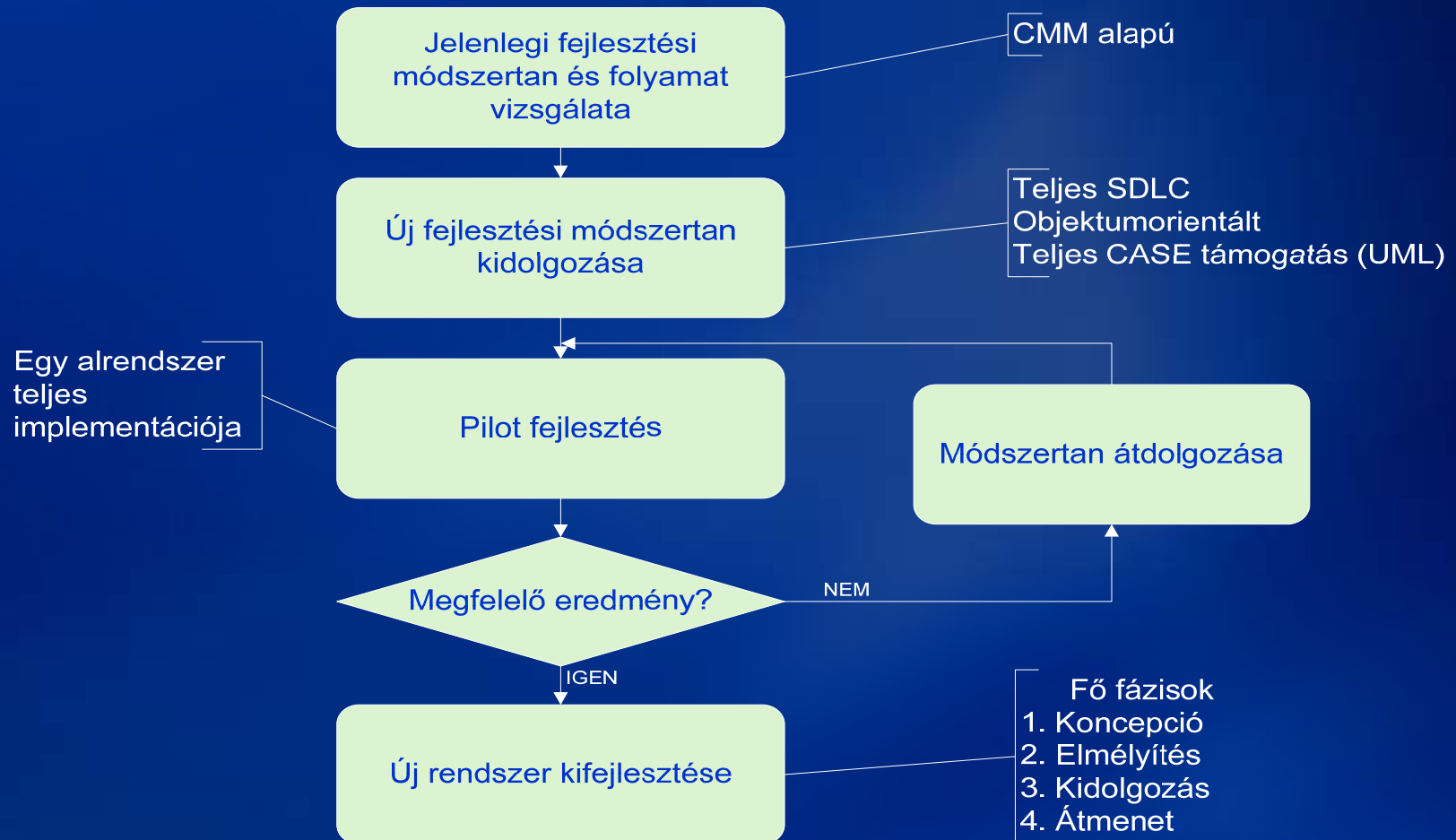
# Esettanulmány: biztonság és SDLC

- **Pénzügyi szektor**
- **Nagyméretű „legacy” rendszer**
  - belső fejlesztés eredménye
  - tranzakció-feldolgozás
  - üzletileg kritikus (a bevételek 80%-át generálja)
  - magas biztonsági követelmények
- **Fő problémák**
  - a változtatásokat nagyon nehéz az elvárt konfidencia szinten implementálni
  - monolitikus architektúra
  - drága
  - néhány fejlesztőtől való szokatlanul nagy függőség

# Esettanulmány: biztonság és SDLC

- **Döntés: teljesen új rendszer fejlesztése**
  - új architektúra
  - új fejlesztési módszertan
  - külső interfészek változatlanok
- **Nehézségek**
  - hiányos szakértelem (az új területeken)
  - magas biztonsági szint fenntartása

# A megközelítés magas szinten



# Biztonsági állapot felmérése

Terület	Státus	Megjegyzés
Rendszerbiztonság	?	Évek óta nem fordult elő érdemi biztonsági esemény. De mit jelent ez?
Rendszerfejlesztés biztonsága		Hiányzik a független kontroll. Nincs biztonsági tesztelés.
Üzemeltetés biztonsága		Példaértékű

## Következtetések:

- Az SDLC integráltan kell kezelje a biztonsági követelményeket
- Biztonsági SDLC (Security SDLC) is szükséges

# Hogyan értünk célt?

- **Gyengeségek**

- a biztonság élelciklus-szemléletének hiánya
- rendszerbiztonság tervezési kompetencia hiánya
- kevésbé boldog fejlesztők

- **Erősségek**

- nagyon jó üzemeltetés-biztonság
- eltökélt felső vezetés

# Hogyan értünk célt? – Az SDLC biztonsági aspektusai

- **Fókuszban a változtatás-kezelés**

- biztonságilag megerősített változtatás-kezelő rendszer
- nem a fejlesztő csapat által kontrollált
- automatikus fordítások és diff-ek
- peer review-k
- részletes naplózás
- rendszeres napló-kiértékelés

- **Betartatott kódolási konvenciók**

- hangsúly a biztonságos kódolás szabályain
- következetes névkonvenciók

# Hogyan értünk célt? – Az SDLC biztonsági aspektusai

- **Szoftver minőségbiztosítási funkció létrehozása**
- **Teszt adatok „mosása”**
- **Ösztönző rendszer kidolgozása**

# Hogyan értünk célt? – Biztonsági SDLC kidolgozása

- **Hatókör:**

- biztonsági komponensek fejlesztése és implementációja (hardver és szoftver egyaránt)
- biztonságilag megerősített infrastruktúra
- biztonsági folyamatok és kontrollkörnyezet

- **Lépések**

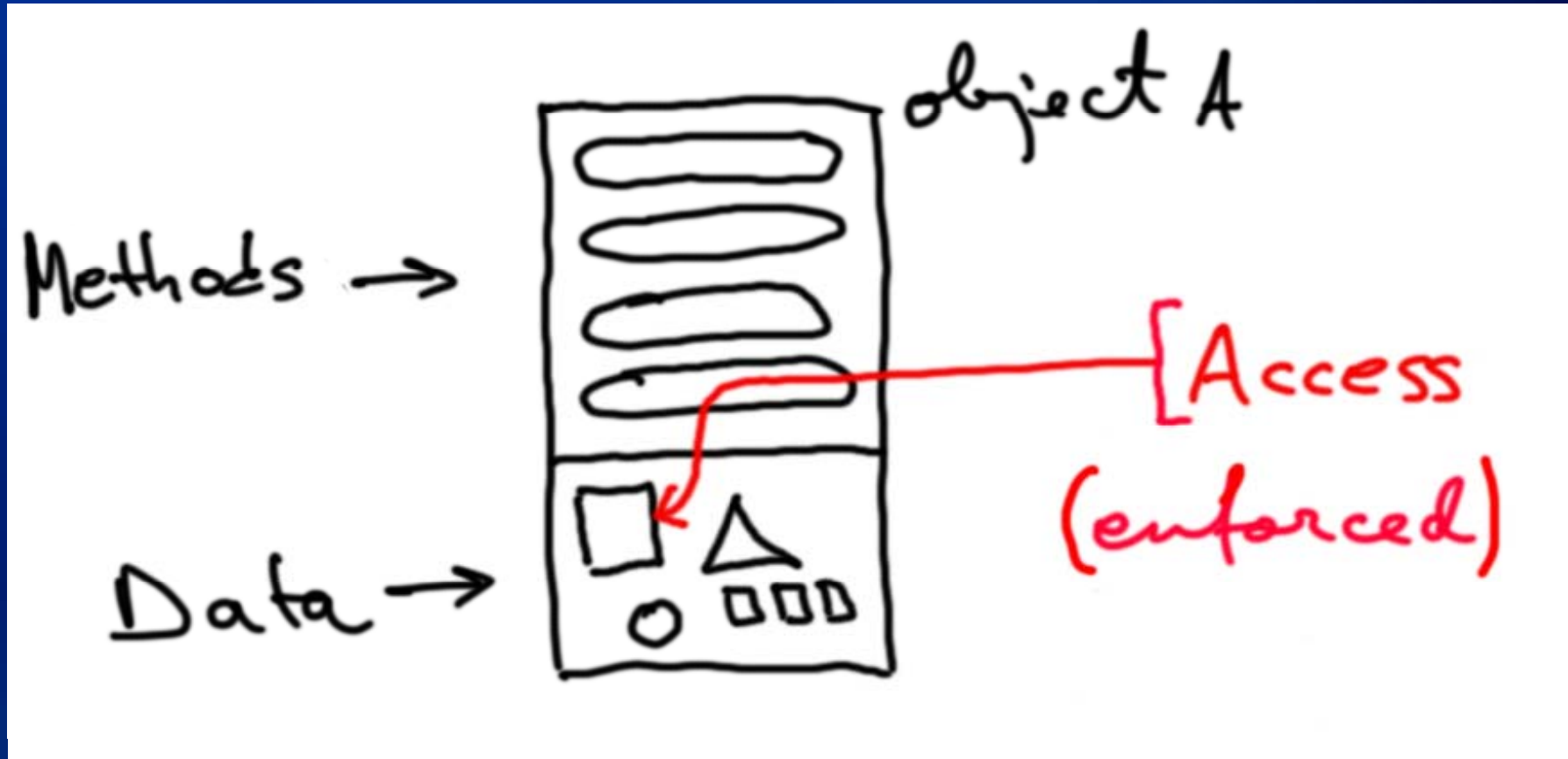
- létező modellek értékelése
- forgatókönyv alapú elemzés
- kiválasztás és testre szabás

- **Szervezeti változtatásokat tett szükségessé**

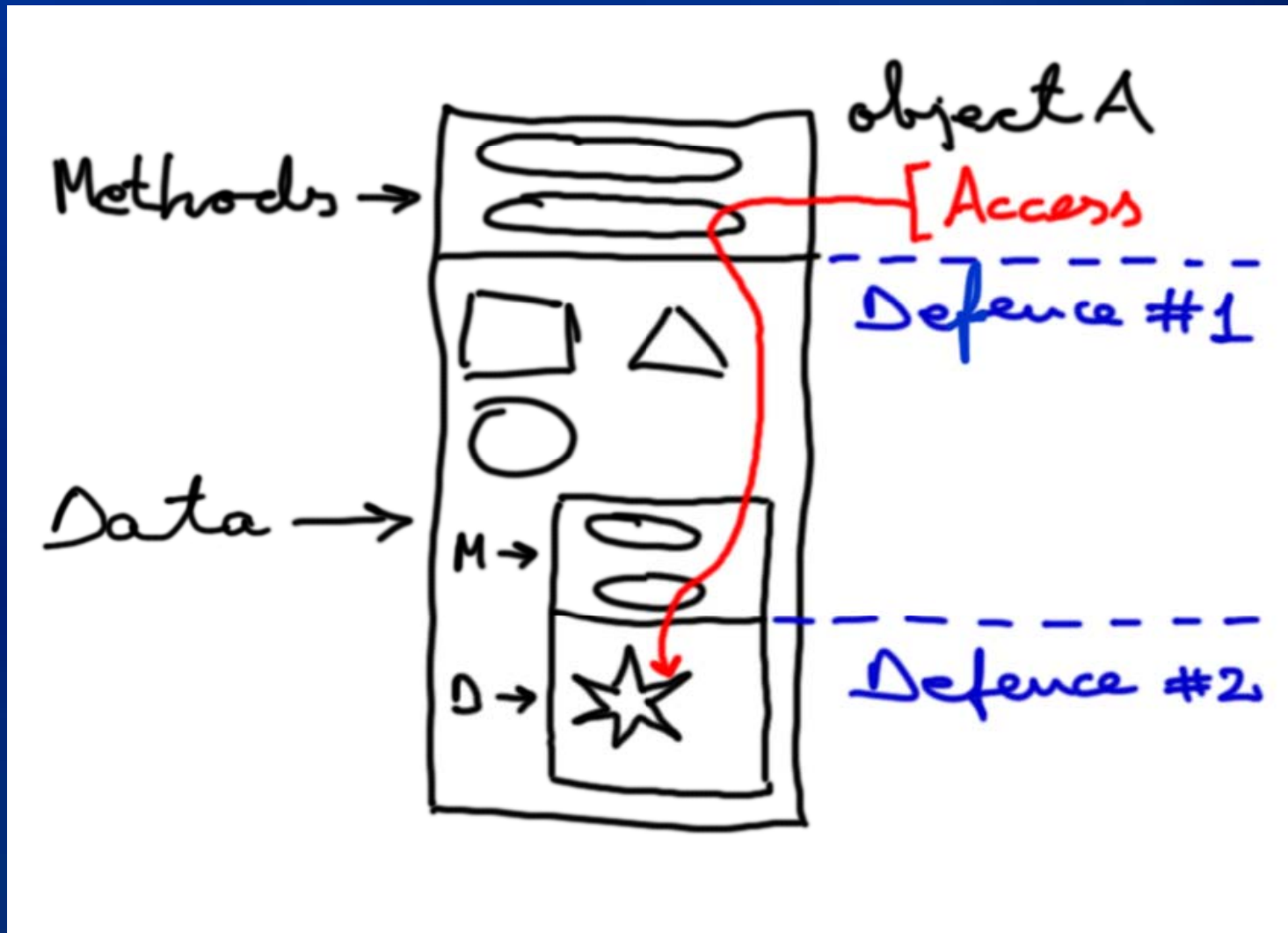
# Objektumorientált és biztonsági tervezési elvek

Biztonsági tervezés	OO tervezés
Mélyégi védekezés	Egységbezárás, öröklés
Biztonságos hibaállapot	Erős kohézió, kivételkezelés
Legkisebb privilégium	Egységbezárás, nyilvános metódusok
Hatáskörök szétválasztása	Gyenge csatolás
Keep it simple	Absztrakció
Teljes mediáció	Egységbezárás, nyilvános metódusok

# Példa: egységbezárás mint teljes mediáció



# Példa: egységbezárás mint mélységi védekezés



# Esettanulmány: mi lett a vége?

- **A „sima” SDLC**

- Viszonylag egyszerű választás az OO paradigma miatt
- Unified Process alapú
- Jelentősen teste szabott
  - követelmény-elemzés egyszerűsítve
  - változtatás-kezelés jelentősen megerősítve
  - tesztelés részletesen szabályozva
  - dokumentációs követelmények, sablonok
  - minőségbiztosítási funkció bevezetése
  - sok gyakorlati útmutató (pl. biztonságos programozási technikák)
- Teljes CASE támogatás
- Sikeres pilot (sok vesződség után)
- Bevezetés, elfogadás

# Esettanulmány: mi lett a vége?

## ● A biztonsági SDLC

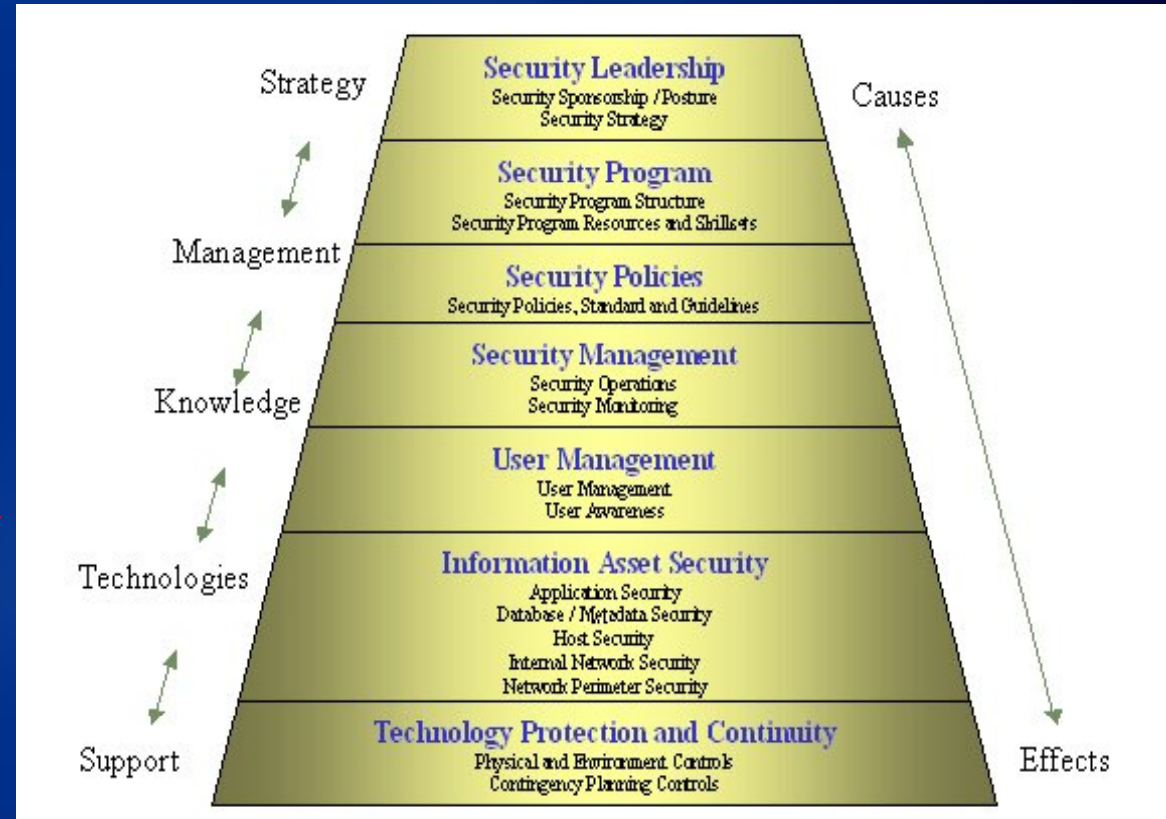
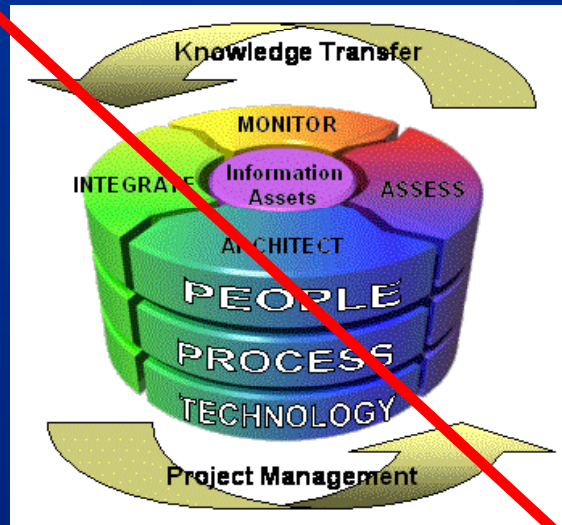
- „staged delivery” inkrementális modell az infrastruktúrára
  - nagyágyúnak bizonyult, jól belebuktunk, ejteni kellett
- PDCA (ISO 27001) a folyamatokra
  - részletes szabályozás mind a négy fázisra
  - viszonylag kis vesződséggel működőképessé tettük
- szervezeti változtatásokra volt szükség
  - Információbiztonsági Osztály függetlenítése
  - IT audit képesség erősítése

# Apropó nagyágyú: tudhattuk volna...

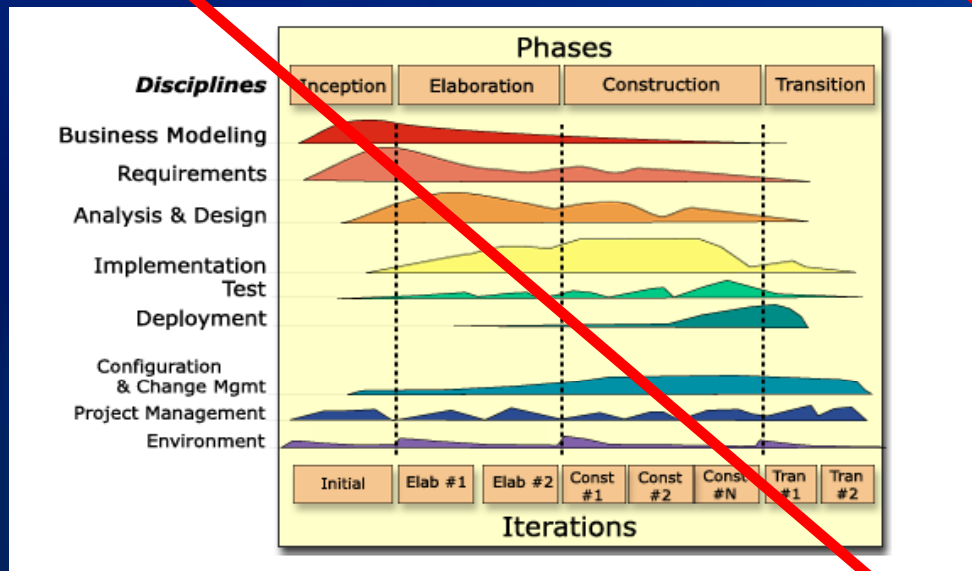
---

# KPMG információbiztonsági módszertan ISS v2.0

Életciklus modell



Képesség modell



Folyamat modell

Unified Process (OO)  
alapú

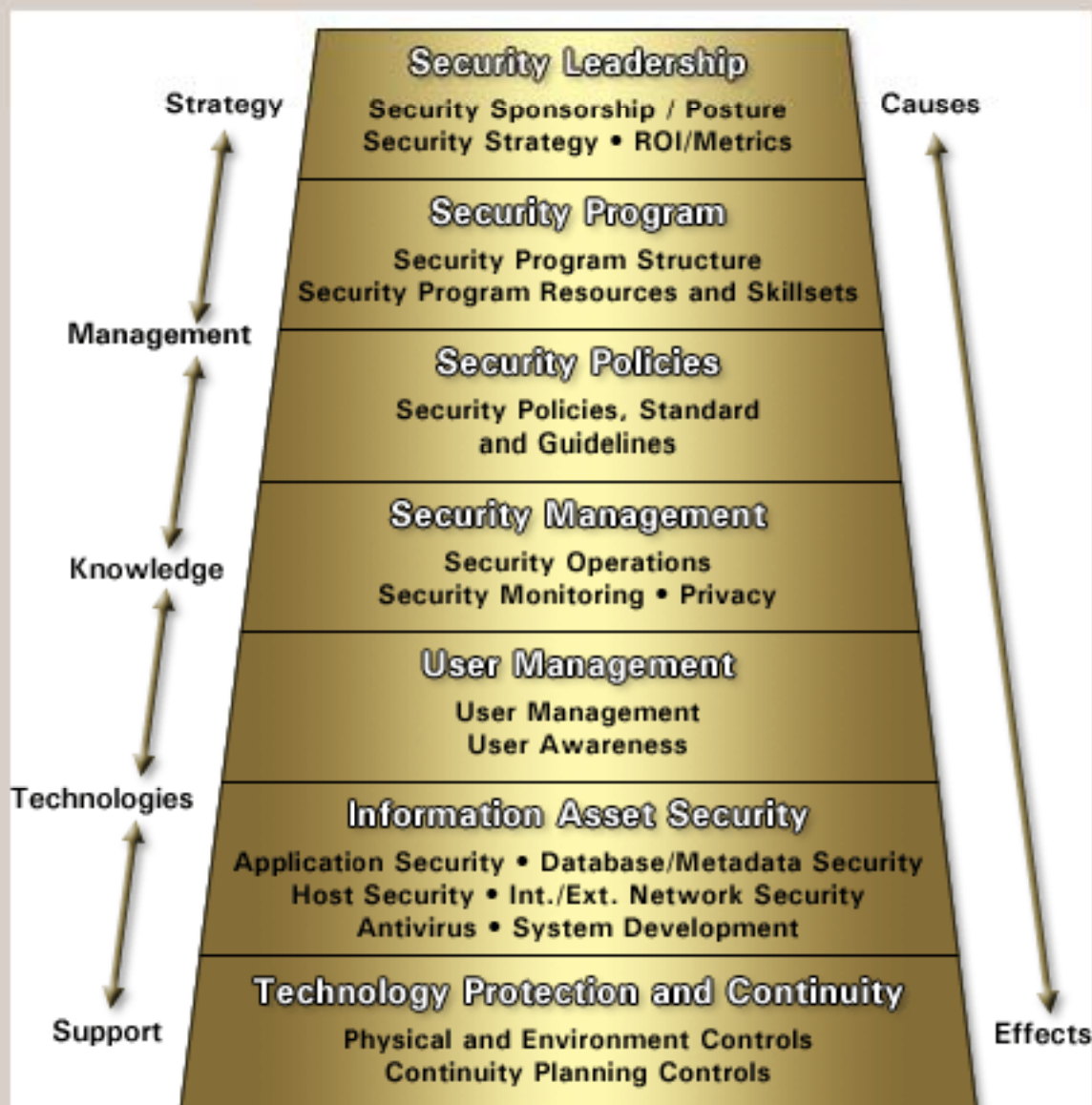


## Content

close all - expand all

- [-] [Security Leadership](#)
  - [Security Sponsorship](#)
  - [-] [Security Strategy](#)
    - [Security Migration Planning](#)
    - [Clients Strategy](#)
    - [ISS Strategy](#)
  - [Metrics / ROI](#)
- [-] [Security Program](#)
  - [-] [Security Program Structure](#)
    - [Security Assessments](#)
    - [Criticality Assessment](#)
    - [Security Organisation](#)
  - [-] [Security Program Resources and Skillsets](#)
    - [Roles and Responsibilities](#)
- [-] [Security Policies](#)
  - [-] [Standards & Guidelines](#)
    - [ISO 17799](#)
    - [Cobit](#)
    - [ISF](#)
    - [ITIL](#)
    - [General Standards & Guidelines](#)
  - [Policies](#)
- [-] [Security Management](#)

## Capabilities Model



# Összegzés

- Nagy biztonságú rendszerek fejlesztése nehezen sikerülhet egy biztonságra kihegyezett SDLC nélkül
- A biztonsági rendszerek fejlesztésében segíthet egy speciális SDLC, de az út rázós
- Érdekes analógia a biztonsági tervezés és az objektumorientált tervezés elvei közt
- A PDCA modell (ISO 27001) jól használható
- A tanácsadók is tévednek



## Az előadó elérhetősége:

**Gaidosch Tamás**

**KPMG**

**+36 (1) 887 7139**

**tamas.gaidosch@kpmg.hu**

**www.kpmg.hu**

The information contained herein [or insert the title of the presentation, report, or talkbook] is of a general nature and is not intended to address the circumstances of any particular individual or entity. Although we endeavor to provide accurate and timely information, there can be no guarantee that such information is accurate as of the date it is received or that it will continue to be accurate in the future. No one should act on such information without appropriate professional advice after a thorough examination of the particular situation.