

CASE technológia

Takács Gábor

CASE:

Computer-Aided Software Engineering

Számítógéppel támogatott szoftverfejlesztés

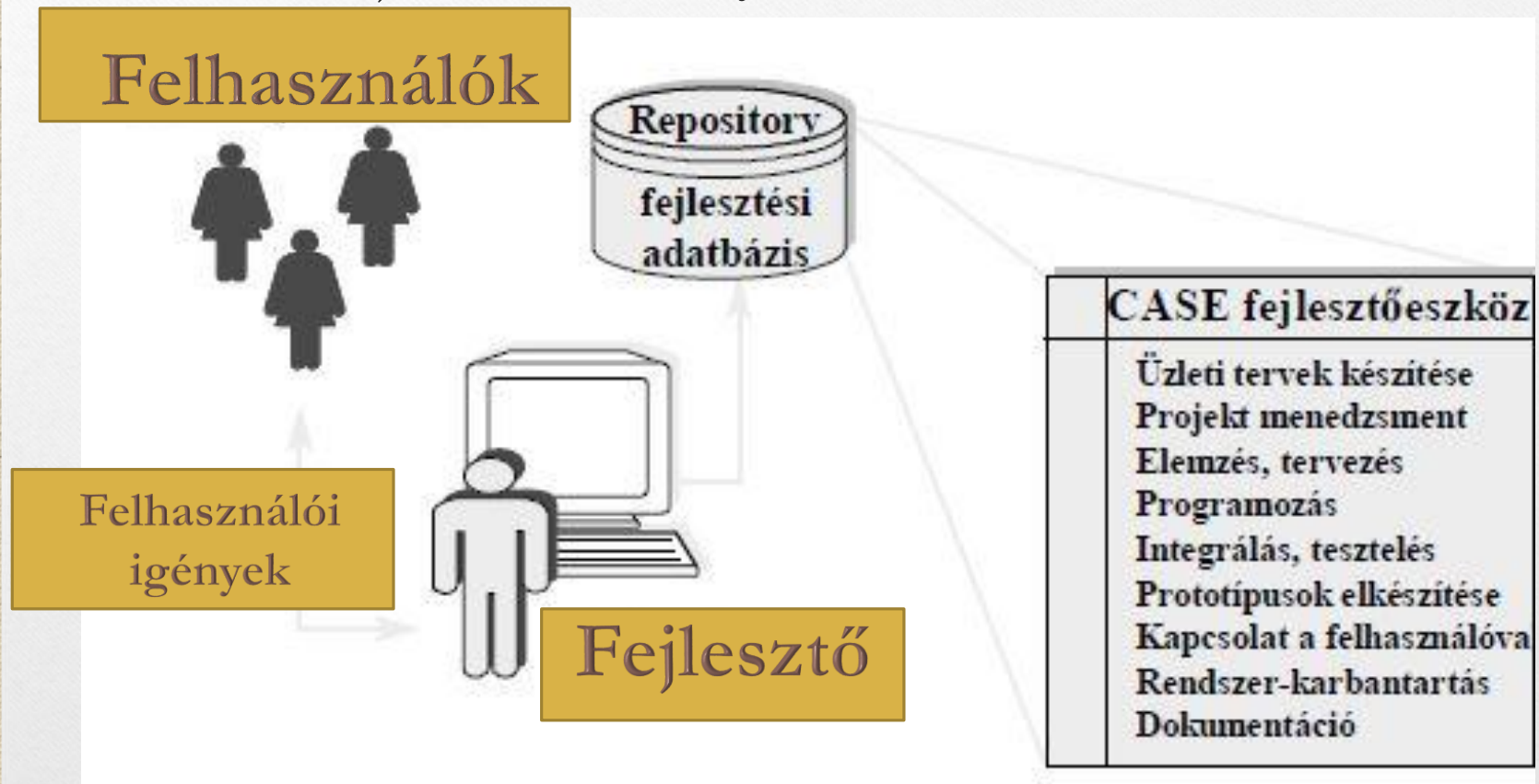
- értelmezés, definíció
- CASE-eszközök működése
- CASE szerepe a fejlesztésben
- kategóriák, szolgáltatások
- szoftverfejlesztés automatizálása
- Komponensek
- fejlesztési adatbázis – repository
- Hatékonyság
- strukturált és objektumorientált CASE-eszközök
- az adott félévben oktatott eszköz, irodalom

CASE definíció

A CASE-eszköz olyan szoftver-támogatás, amely nemcsak eszközkészletet, de **módszereket is biztosít** az

- informatikai fejlesztési folyamat, magas színvonalú, hatékony végzéséhez,
- fejlesztési hibák minimalizálásához és
- a fejlesztett alkalmazások karbantartásához, változtatásához azzal, hogy
- segít a rendszer komponenseinek és működésének feltárásában
- ellenőrzi és tárolja a fejlesztési információkat,
- irányítja a fejlesztési projektet és
- adminisztrálja a fejlesztéshez kapcsolódó információkat és teendőket.

A számítógéppel támogatott fejlesztés folyamata és elemei



A CASE elnevezés

Elnevezés a **Nastec Corporation of Southfield-től (Michigan)** származik **1982-ből**, amikor is az általuk mikroszámítógépre kifejlesztett GraphiText integrált grafikus szövegszerkesztőjüket alkalmassá tették a szövegben való keresésre képes hyperlink kapcsolatok kezelésére.

CASE-eszközök osztályozása

1. Az *eszközök* (Tools) kategória csupán néhány speciális feladat végrehajtását támogatja
2. A *munkafolyamat* (Workbench) típusú eszközök a fejlesztési életciklus egyes fázisainak a végrehajtását támogatják, míg
3. A *fejlesztési környezet* jellegű (Environments support) CASE-ek a teljes életciklushoz kínálnak automatikus megoldásokat

1. CASE-eszközök→Tools

Ezen eszközök számos megoldást kínálnak a fejlesztési munkák automatizálására, így pl.

- a funkcionális követelmények vizualizálására,
- prototípusok egyszerű és gyors előállítására,
- user interface-ek tervezésére
- A funkcionális követelmények tervmodellé transzformálására

1. CASE-eszközök→Tools

Megkülönböztethető csoportjaik:

1. Életciklus-támogatás (Life-cycle support) →

- Upper CASE (stratégiai tervezés, projektvezetés) stratégiai és logikai tervek készítése hagyományos technikákkal:
- Lower CASE (egyszerűbb specifikációk) az életciklus második részét, a fizikai tervezést és a megvalósítást támogatják
- Middle CASE (elemzés és tervezés)

2. Integráció szintje szerinti kategóriák (Integration dimension)

- CASE Framework
- ICASE Tools
- Integrated Project Support Environment (IPSE)
- Construction dimension

2. Munkafolyamat-támogatás, azaz Workbench

Speciális feladatok támogatását végzik, így

- segítik a homogén (azonos elemekből álló) és konzistens (belső ellentmondásoktól mentes) interface-tervezést.
- egyszerűen kezelhetők és a tervmodellekbe beágyazhatók (control integration).

2. Munkafolyamat-támogatás, azaz Workbench

A CASE workbench-ek alkalmazási területei:

- Üzleti domén modellek tervezése
- Elemzési, tervezési feladatok segítése
- Felhasználói interfészek tervezése
- Programozás
- Verifikációs és validitási tesztelés támogatása
- Karbantartás, visszafejtéses fejlesztés
- Konfigurációs menedzsment
- Projektmenedzsment

3. Fejlesztési környezet, azaz Environments support

A Fejlesztési környezet támogatású rendszerek egyesítik a CASE tool-ok és a workbench-ek képességeit.

Az alábbi csoportokat különböztetjük meg:

- Toolkit-ek
- Nyelv-orientált eszközök → Language-centered tools
- Integrált eszközök
- Negyedik generációs megoldások → 4GL
- Folyamatorientált támogatás → Process-centered

CASE szolgáltatások

- a fejlesztési folyamat adott módszertan szerinti irányítása
- fejlesztési eszközök integrálása
- vizualizáció, diagramok készítése, fejlesztési adatok ellenőrzése
- fejlesztési adatbank létrehozása, kezelése
- a fejlesztési fázisok eredményeinek összehangolása
- jelentéskészítés
- a rendszer logikai helyességének ellenőrzése
- fejlesztési dokumentáció készítése
- projektmunka irányítás, verziókezelés

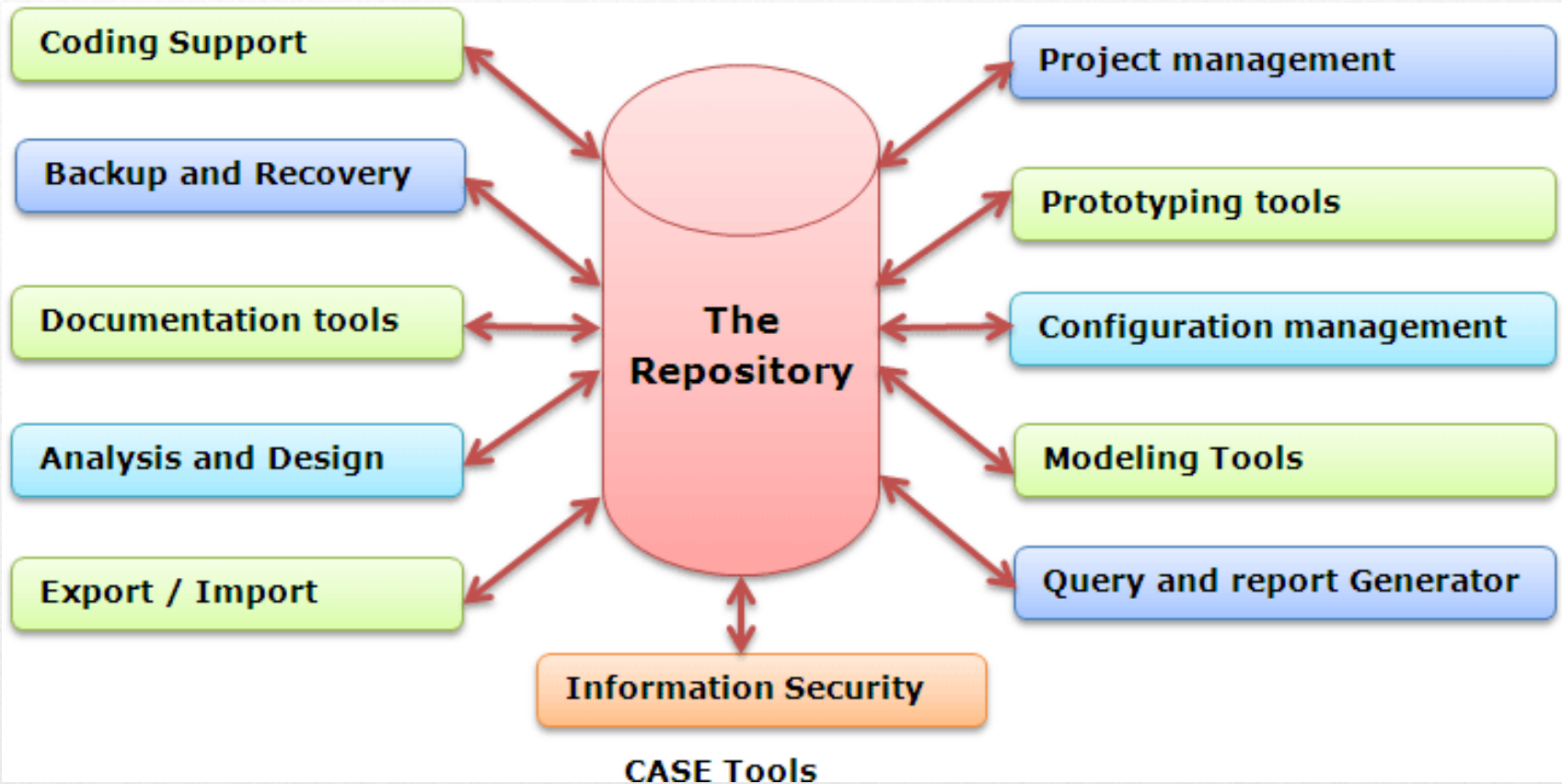
Repository

Repository: A fejlesztési adatbázis

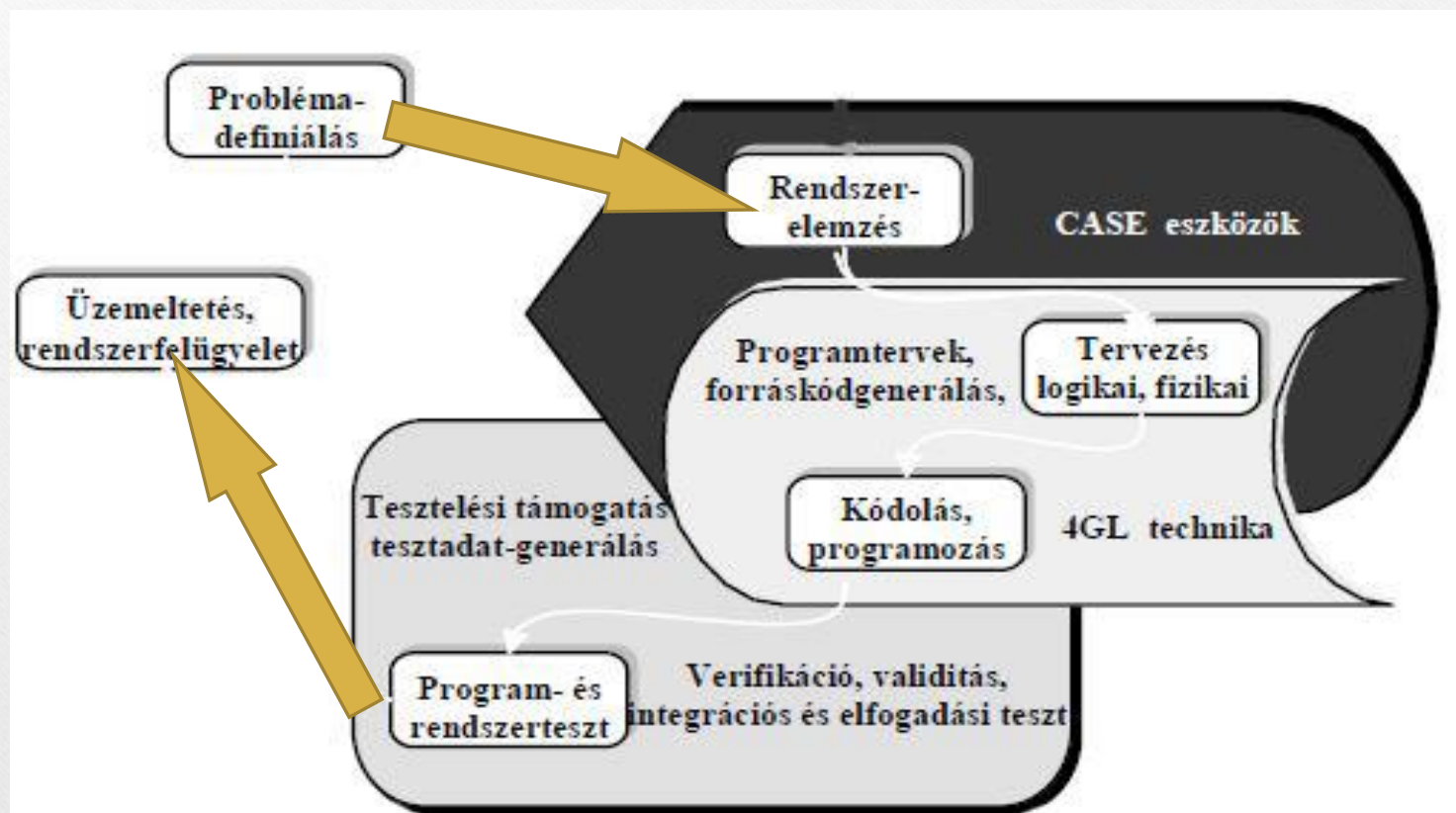
A fejlesztési adatbázisban minden olyan információ megtalálható, amely a fejlesztési munka során keletkezik, így a :

- megoldandó probléma és annak sajátosságai,
- a jelen rendszer modelljére vonatkozó jellemzők: elemek, azok jellemzői, specifikációja, kapcsolatok
- szabályok és elvárások, amelyeket a fejlesztés során követni kell.

CASE szolgáltatások



A CASE helye a szoftverfejlesztésben



A fejlesztőeszközök hatékonysága

- a munka hatékonyságának a növelése
- fejlesztési idő radikális csökkentése
- függetlenség biztosítása:
 - Platformfüggetlenség
 - hálózati függetlenség
 - adat(bázis)függetlenség

CASE eszközök

UML-alapú CASE-eszközök:

- http://en.wikipedia.org/wiki/List_of_Unified_Modeling_Language_tools

Bemutatóra kerülő eszközök:

- WinDev: <http://www.windev.com/>

WinDev

Az integrált fejlesztő környezetű (IDE) 4GL támogatású WinDev-et a PCSoft cég fejlesztette ki 1993-ban eredetileg azzal a céllal, hogy megkönnyítsék az adatközpontú Windows-alkalmazások készítését.

A WinDev procedurális és objektumorientált szemléletű eszköz, amely egyaránt támogatja a Java és a .Net platformokon való fejlesztéseket.

Az eszköz számos előredefiniált szabványos formalizmusa és algoritmusai megkönnyítik, **sőt automatizálják a programozási munkát**, lehetővé téve ezáltal a gyors alkalmazásfejlesztést (RAD). A forráskód előrefordított és futási időben interpretált, így a végrehajtható program operációsrendszer-független

WinDev

A 16 verziótól a WinDev már Linux alkalmazások írására is használható.

A kapcsolódó termékek:

- **WebDev** (weboldalak és webalkalmazások fejlesztését támogatja) és a
- **WinDev Mobile** (mobil-eszközök → Smartphonok, Pocket PC-k, ipari eszközök stb.) ugyanazokat az elveket követik mint a WinDev, és ugyanazt a programnyelvet használják (WLanguage).

Előnyük, hogy a fejlesztések különböző platformokon is működőképesek.

A WinDev saját integrált adatbázis-kezelő megoldásokat alkalmaz, de támogatja az általánosan használtakat is.

923
NEW FEATURES

What's New in WINDEV 23 ▶

DEVELOP 10 TIMES FASTER

WINDEV.

Windows & .Net
Development



Create 10 times faster all your applications
for Windows, .Net, Linux and Mac

WEBDEV.

WEB & Cloud
Development



Create Responsive Web Design sites
10 times faster

**WINDEV.
Mobile**

Android & iPhone
Development



Create in just a few hours your
iOS, Android, Windows 10 Mobile
applications

WINDEV seminar



WINDEV seminar



Alkalmazás 3 perc alatt (iPhone)



ELMÉLET

féléves feladat

teljesítéséhez

PROJEKT

- A **projekt** egy olyan egyszeri feladat, amelyet adott idő és erőforrások felhasználásával az erre a feladatra szervezett munkacsoport hajt végre.
- Tehát sohasem rutinmunka.

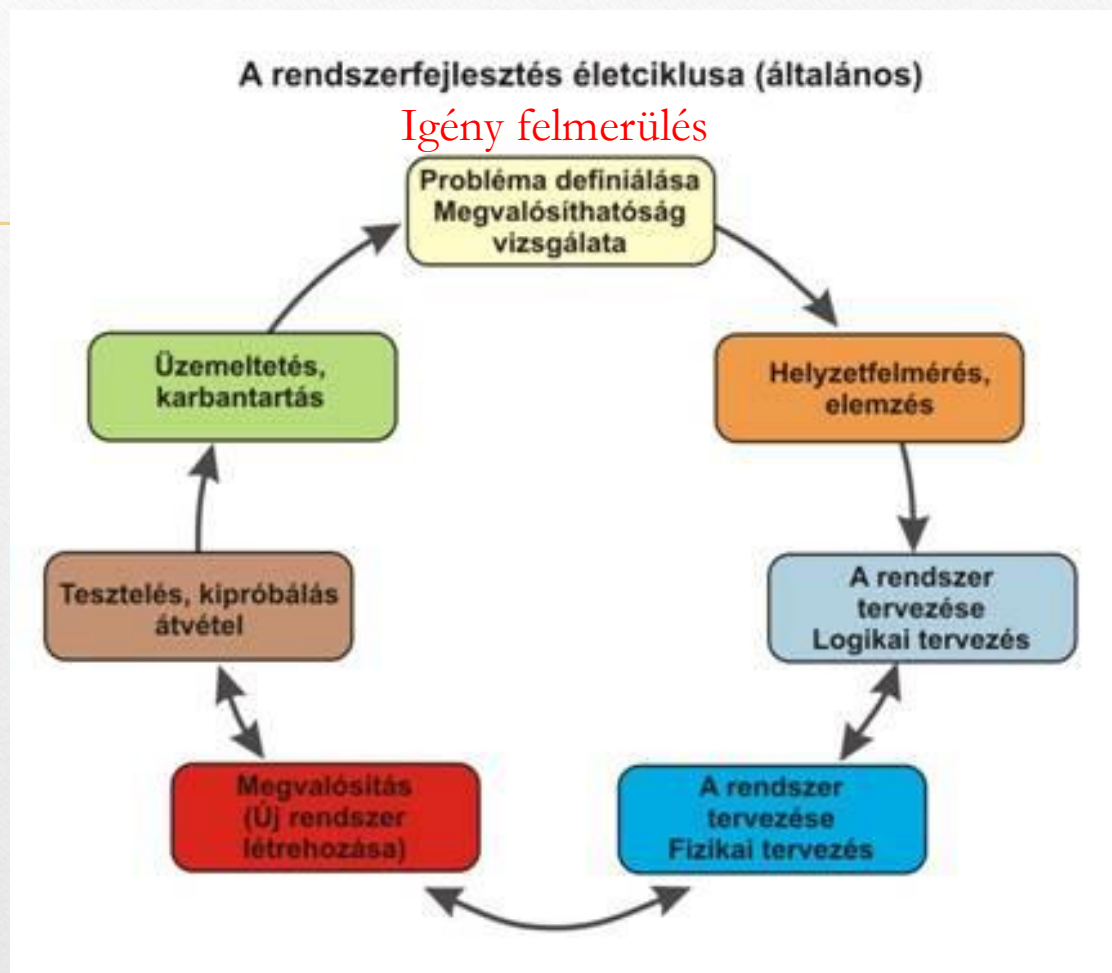
Rendszerint sok, különböző szakmabeli résztvevő munkáját kell összehangolni. A projektben egy **meghatározott cél** végrehajtása érdekében egymáshoz kapcsolódó tevékenységeket végeznek a résztvevők.

A projekt során minden szereplőnek egyértelműen meghatározott feladata van, ezek ellenőrzését végzik a projektirányítási, és minősegbiztosítási szakemberek.

IR fejlesztés életciklusa,

Életciklus : Az igény
felmerülésétől a termék
használatból való
kivonásáig
(feledésbe merülésig)
terjedő időtartam.

Egy PC alapú
információs **rendszer**
életciklusa
(system life cycle)
négy szakaszból áll:
születés, fejlődés, termelés
és halál.



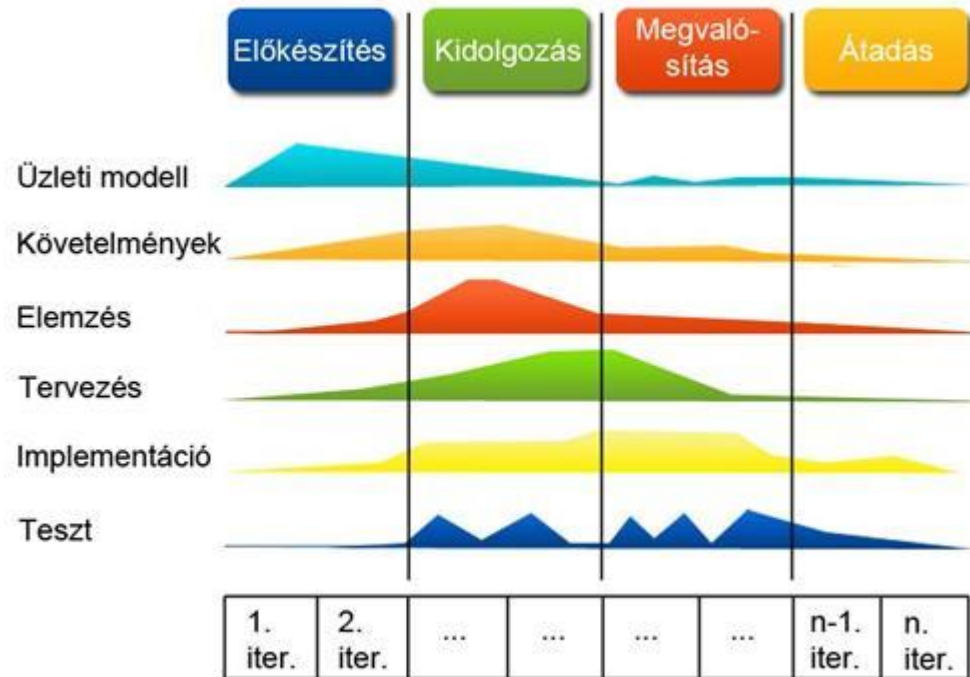
IR fejlesztés életciklus fázisai

Információrendszer-fejlesztési tevékenység

- több szakember, hosszabb munkafolyamata

Felosztás:

- fázisok
- mérföldkövek



ELŐKÉSZÜLETI, ELEMZÉSI FÁZIS

Cél: *probléma meghatározása*

Első lépés:

meg kell fogalmazni a felismert problémát,
problématerületet, a megoldandó feladatot.

- problémajelenség megfogalmazás
- problémával kapcsolatos ismeretek gyűjtése
- problémák meghatározása munkahelyenként, és szerepkörönként
- ok-okozati összefüggések keresése
- lehetséges megoldások alkalmazása

Második lépés:

fejlesztendő problémakörnyezet elemeinek és a rendszer
működésének feltárása, tények rögzítése (Is Modell, Van modell)

ELŐKÉSZÜLETI, ELEMZÉSI FÁZIS

Problémaelemzés technikai eszközei (számos technika áll rendelkezésre):

- SWOT- analízis (felsorolás)
elemek csoportosíthatósága miatt fontos, a problémák rendszerre gyakorolt hatását vizsgáljuk. **Ne mindent soroljunk fel**, hanem azokat, amik a cél szempontjából meghatározóak, és befolyásolják a problémamegoldást.

SWOT- elemzés	SEGÍTIK a célok elérését	GÁTOLJÁK a célok elérését
BELSŐ TÉNYEZŐK (szervezeti jellemzők)	ERŐSSÉGEK (strengths)	GYENGESÉGEK (weaknesses)
KÜLSŐ TÉNYEZŐK (környezeti jellemzők)	LEHETŐSÉGEK (opportunities)	FENYEGETETTSÉGEK (threats)

ELŐKÉSZÜLETI, ELEMZÉSI FÁZIS

- **Ishikawa-diagram** (halszálka-diagram)

okok és okozatok összefüggéseit vizsgálja

Először magát a problémát rögzítjük, majd számba vesszük a lehetséges okokat.



ELŐKÉSZÜLETI, ELEMZÉSI FÁZIS

A diagram kialakításában elsősorban a 4M vagy 5M struktúra kialakítására kell törekedni, amely szerint a 4, illetve 5 fő ok csoport a következő:

- ember (Man);
- gép (Machine);
- anyag (Material);
- módszer (Method);
- mérés (Measuring).

ELŐKÉSZÜLETI, ELEMZÉSI FÁZIS

- **Pareto-analízis:**

problémát okozó tényezők súlyosságát, és hatásának mértékét vizsgáljuk.

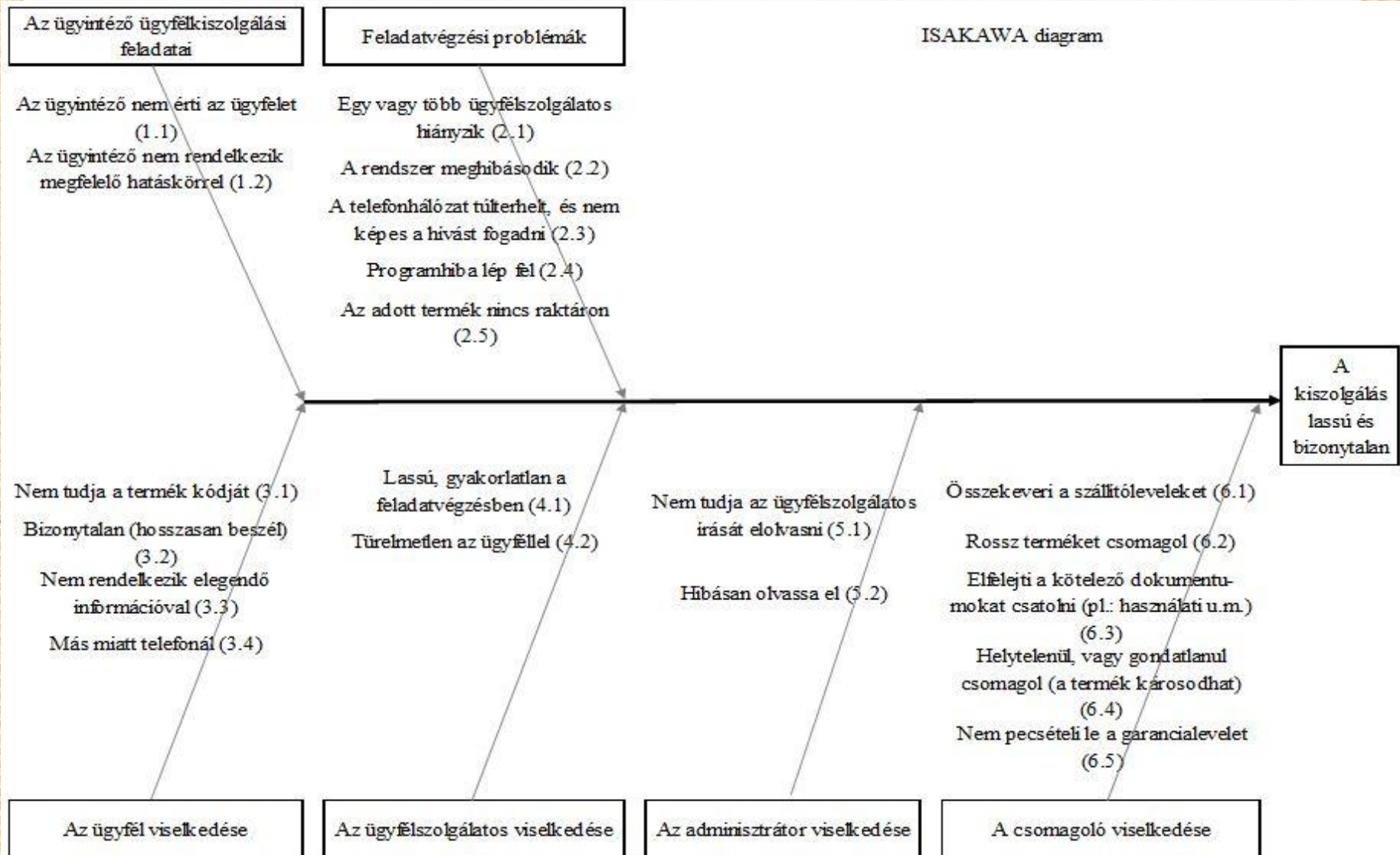
Probléma okok előfordulási gyakoriságai

=

fontossági sorrend

Hangsúly: a leggyakrabban előforduló,
legkritikusabb tényezőkön kell legyen!

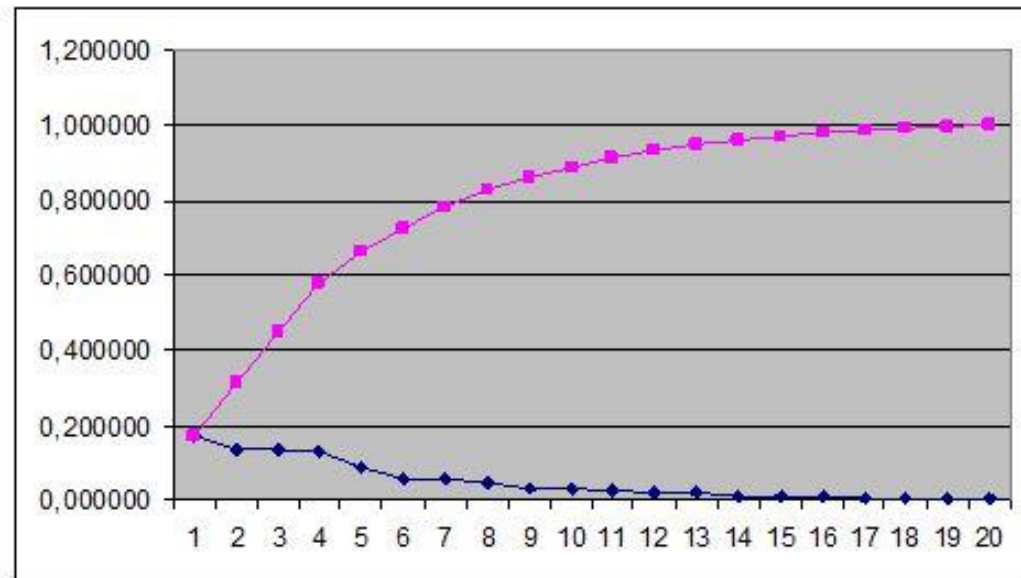
Példa egy feladatra



Példa egy feladatra

PARETO diagram

A problémák felmerülési gyakorisága				
Probléma azonosító		Előfordulási gyakoriság	Relatív gyakoriság	Halmazott gyakoriság
1	2,3	36	0.174757	0.174757
2	3,3	28	0.135922	0.310680
3	5,1	28	0.135922	0.446602
4	3,2	27	0.131068	0.577670
5	1,1	18	0.087379	0.665049
6	2,6	12	0.058252	0.723301
7	4,2	12	0.058252	0.781553
8	3,4	10	0.048544	0.830097
9	6,3	6	0.029126	0.859223
10	6,4	6	0.029126	0.888350
11	6,2	5	0.024272	0.912621
12	2,1	4	0.019417	0.932039
13	3,1	4	0.019417	0.951456
14	6,1	2	0.009709	0.961165
15	2,2	2	0.009709	0.970874
16	5,2	2	0.009709	0.980583
17	1,2	1	0.004854	0.985437
18	2,5	1	0.004854	0.990291
19	4,1	1	0.004854	0.995146
20	6,5	1	0.004854	1
Összesen:		206	1	



X: probléma okok

lila: halmazott gyak.

Y: előfordulási gyakoriság

kék: relatív gyak.

Helyzetfeltárás, -elemzés

- Rendszerszemléletből kiindulva kell megkezdni
- Folyamatokat kell elemezni, figyelni...

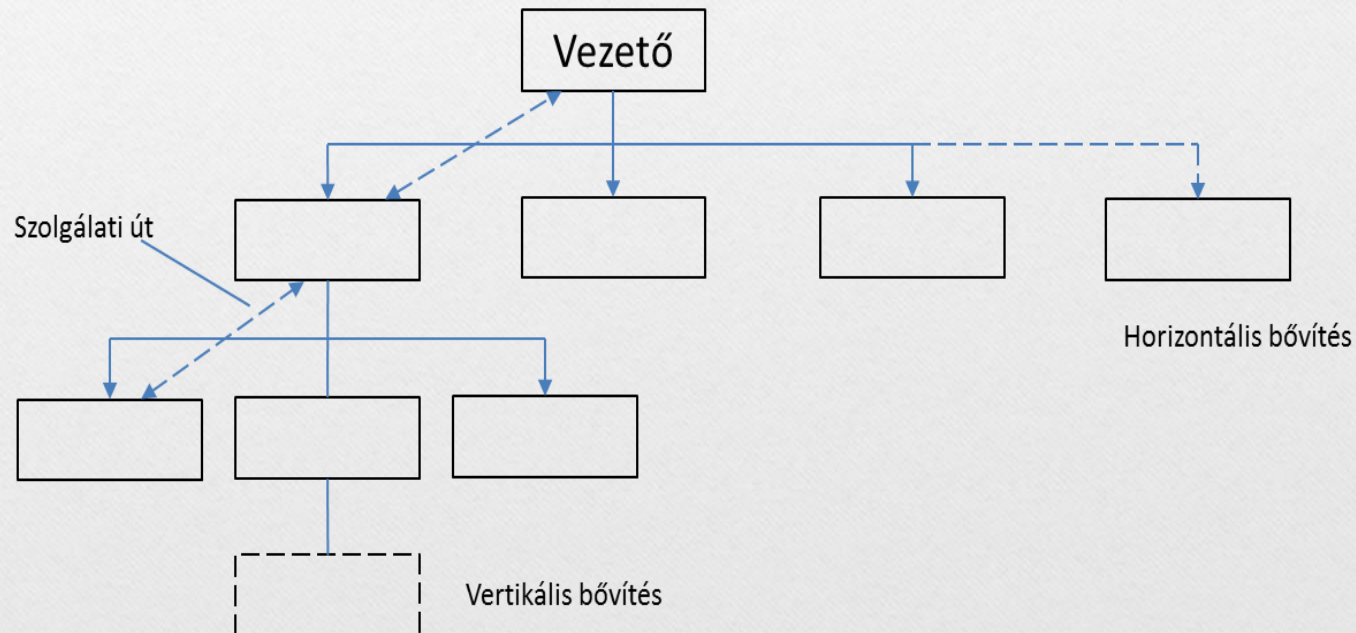
Megismeréshez leggyakrabban alkalmazott megoldások:

- ...
- szervezeti struktúra (horizontális, vertikális)
- Folyamatmodellezés (folyamatábra, folyamat gráf, stb.)

Helyzetfeltárás, -elemzés

Vertikális szervezeti struktúra:

(Szervezetek különböző irányítási szintjeit, és a szervezeti egységek egymáshoz való viszonyát.)



Helyzetfeltárás, -elemzés

Folyamatok:

- **Elem**

Pl.: reggeli készülődés eleme: fogmosás

- **Entitás (tulajdonság)**

fogkefe: rugalmassága, anyaga, sörteszáma, stb.

fogkrém: minősége, íze, állaga, stb...

- **Totalitás**

próbálunk a teljességre törekedni...

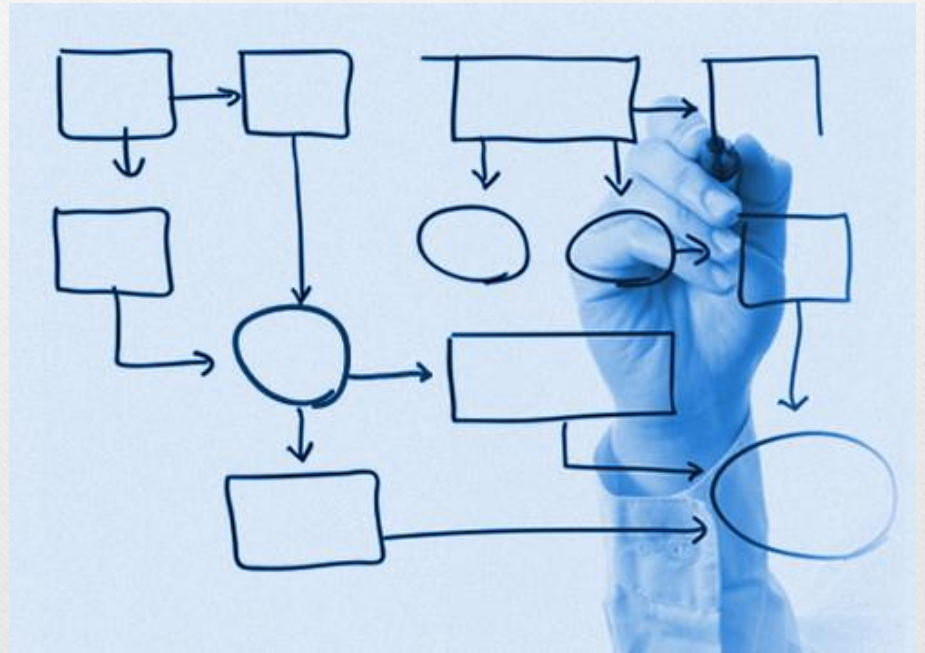
2. fázis: Helyzetfeltárás, -elemzés

Folyamatmodellezési technikák:

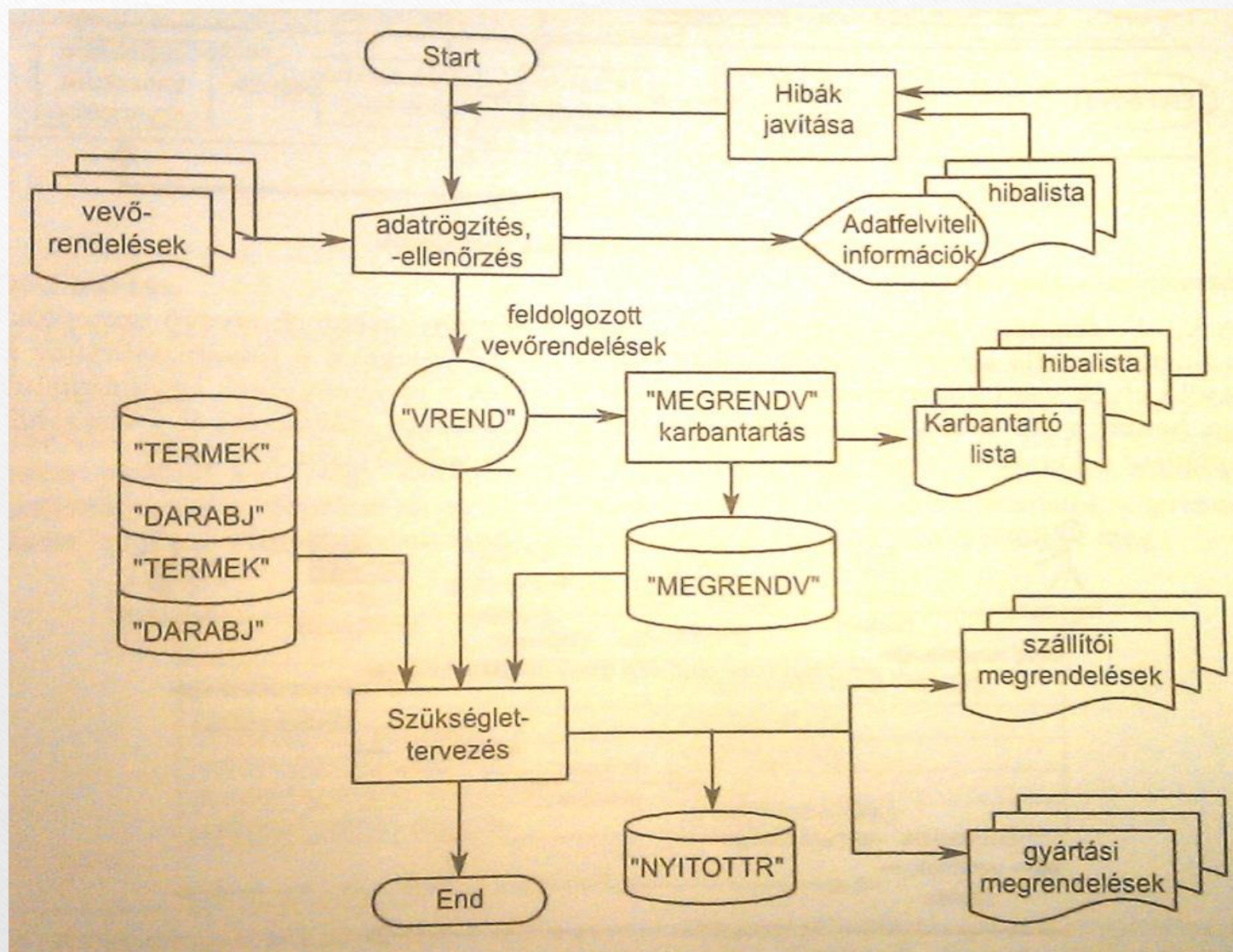
Folyamatábra: bemutassa egy adott folyamat elemeit, feladatokat egymásutániságát

Fontos szabály:

a kapcsolatjelző nyilak csak jobbra,
illetve lefelé mutathatnak.
(fel, bal csak visszacsatolás)



Vevői rendelés példa

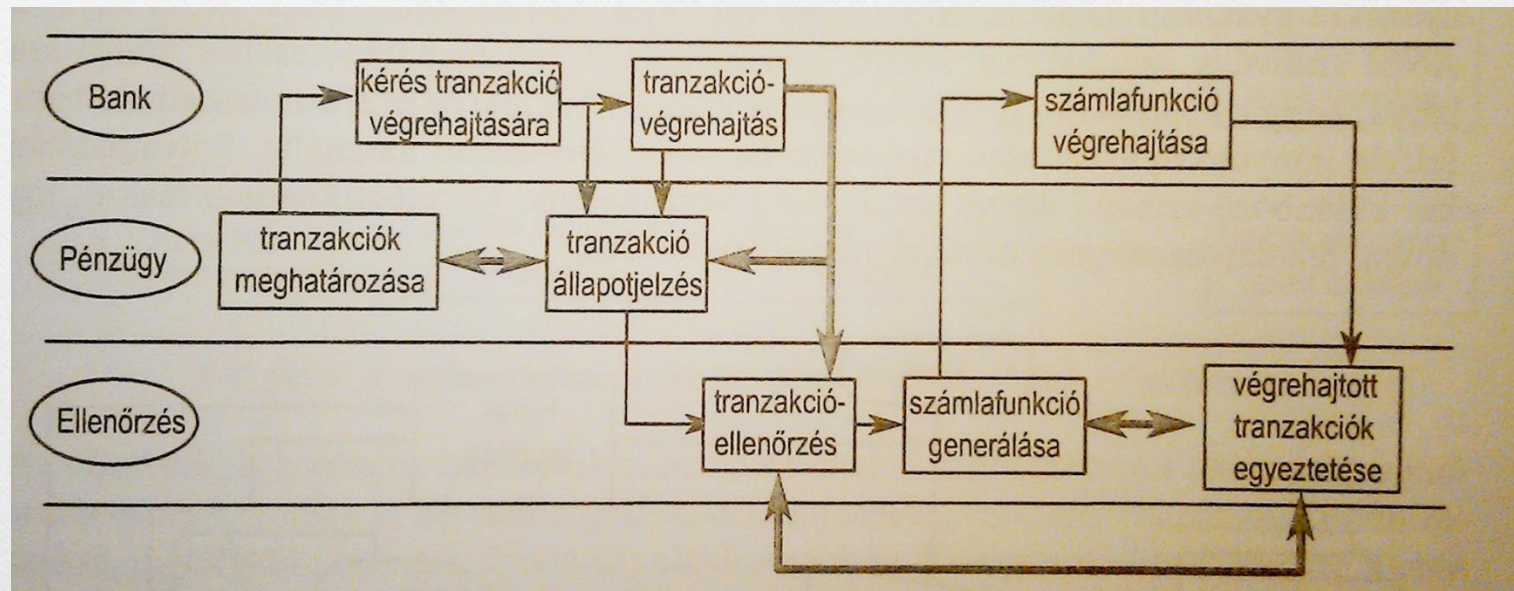


Helyzetfeltárás, -elemzés


Folyamatmodellezési technikák:

Folyamat gráf (P-gráf):


jól szemlélteti (szervezeti, rendszer) egységek által végzendő tevékenységek sorrendje, és egymáshoz való viszonya



On-line megoldás




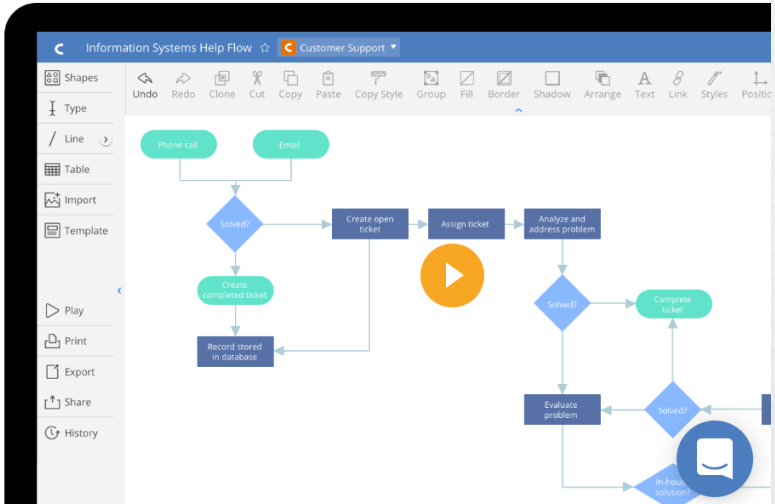
by nulab

FeaturesPricingTemplatesFor teams Try it freeLog in

Cloud-based diagrams, the easy way.

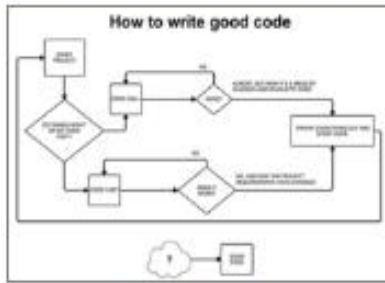
Diagram maker for companies, teams, and you





The screenshot displays the CACOO web application interface. On the left is a sidebar with a 'Shapes' panel containing icons for various diagram elements like rectangles, circles, and lines, as well as a 'Type' panel with options like 'Line', 'Table', 'Import', and 'Template'. The main workspace shows a flowchart titled 'Information Systems Help Flow'. The flow starts with two input shapes: 'Phone call' and 'Email'. These lead to a decision diamond labeled 'Solved?'. If 'No', the flow goes to 'Create open ticket', then 'Assign ticket', and 'Analyze and address problem'. If 'Yes', it goes to 'Create completed ticket'. The 'Create open ticket' path also leads to 'Record stored in database'. The 'Analyze and address problem' path leads to another decision diamond 'Solved?'. If 'No', it goes to 'Evaluate problem', then another decision diamond 'Solved?'. If 'Yes' at either 'Solved?' diamond, the flow goes to 'Complete ticket'. The final step is 'To Home' with a home icon.

Ingyenes, gyors, látványos: <https://cacoo.com/lang/hu/>



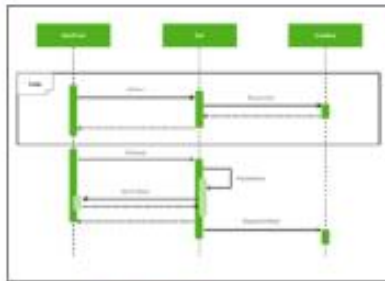
Flowcharts



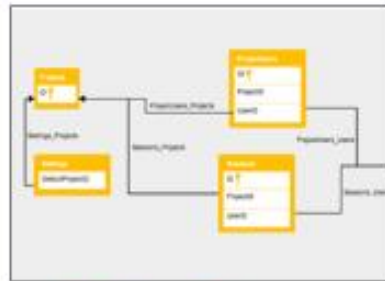
Wireframes & Mockups



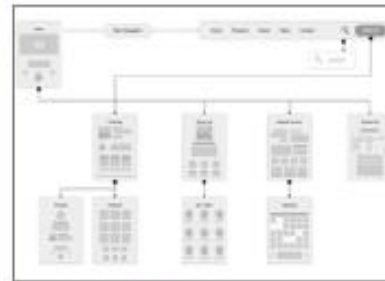
Network Diagrams



UML Diagrams



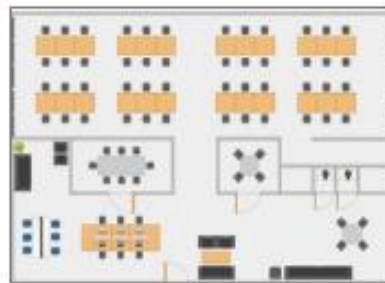
Database Diagrams



Sitemaps



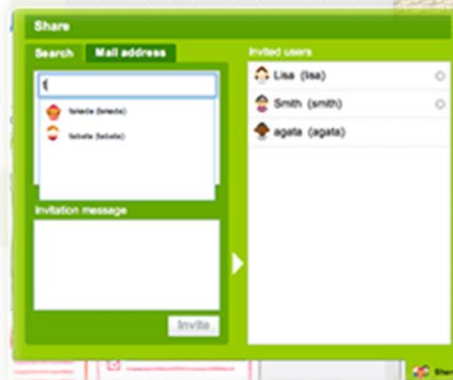
Mind Maps



Room Layouts



Templates



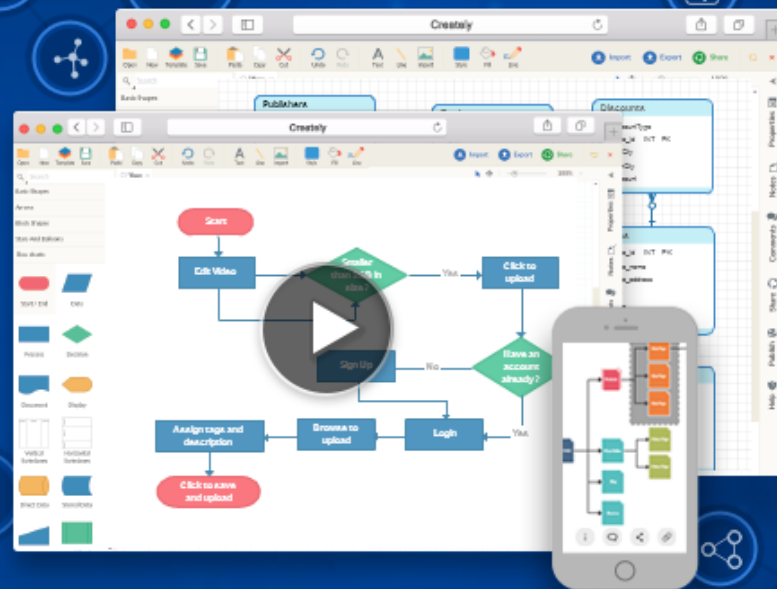
[Features](#)[Products](#) ▾[Examples](#)[Pricing](#)[Community](#)[Resources](#) ▾[About](#) ▾

DIAGRAMMING & COLLABORATION

Beautiful, Easy and built for teams.
Flowcharts, UML, UI Mockups and more.

Try Creately Now

No registration required



trusted
by



NATIONAL
GEOGRAPHIC

PayPal

citi



amazon.com

+ 1000's
more

view
testimonials



About

[How to start](#)[How to use](#)[Get full ARIS](#)[Quick reference](#)[Download for Free now!](#)

Version 2.4 available since Dec 19, 2012

Download your free of charge copy of
ARIS Express right now

Forgatókönyvek

- Követelményspecifikáció:

dokumentum a feladatokról, és meghatározzák a
fejlesztéshez, üzemeltetéshez szükséges erőforrásokat.

- Tartalma:

Specifikációs folyamat során keletkezett tartalmak, tehát:

- szereplők listája
- funkciók részletes leírása
- fentiek együttműködése, folyamatok
- Use Case modellek
- pontosított fogalomtár

-

USE CASE technika

Célja: funkciók ábrázolása úgy, hogy a szereplők, folyamatok, és azok kapcsolatrendszerének és környezetésnek a bemutatása akár elemi szintig lebontva áttekinthetőek legyenek.

Követelményrögzítés: középpontban a rendszer által végrehajtandó funkciók vannak.

Megmondja, hogy

- mit kell tudnia- és,
- milyen funkciói legyenek a rendszernek.

Tulajdonságai:

- szemléletes,
- könnyen áttekinthető

USE CASE technika

A diagram elemei

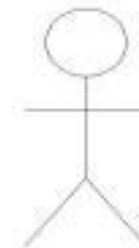
- ◆ **Rendszer (system):** amit el akarunk készíteni



- ◆ **Aktor (actor):** egy szerepkört reprezentál.

Környezet: a világ, ami a rendszert körülveszi → aktorok

A felhasználók akik a rendszert használják



név

- ◆ **Használati esetek (use case):** elvárt viselkedési minták (mire képes a rendszer)



Diagram elemek azonosítása

Kapcsolatok azonosítása: igék és főnevek közti kapcsolatok alapján

Aktorok azonosítása:

főnevek keresése a szöveges specifikációból

- kik rendszer használói ? (user)
- ki felel a rendszer karbantartásáért ?
- mik a rendszer által használt erőforrások ?
- mik a rendszerhez kapcsolódó
más rendszer(ek) ?

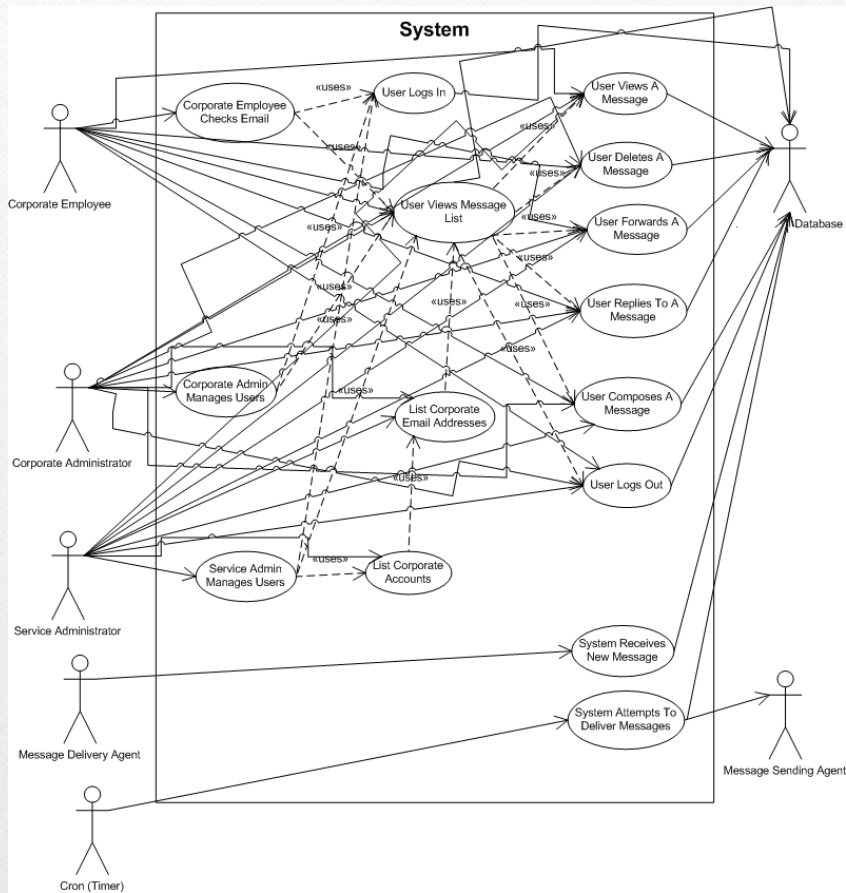
Használati esetek azonosítása:

igék keresése a szöveges specifikációból

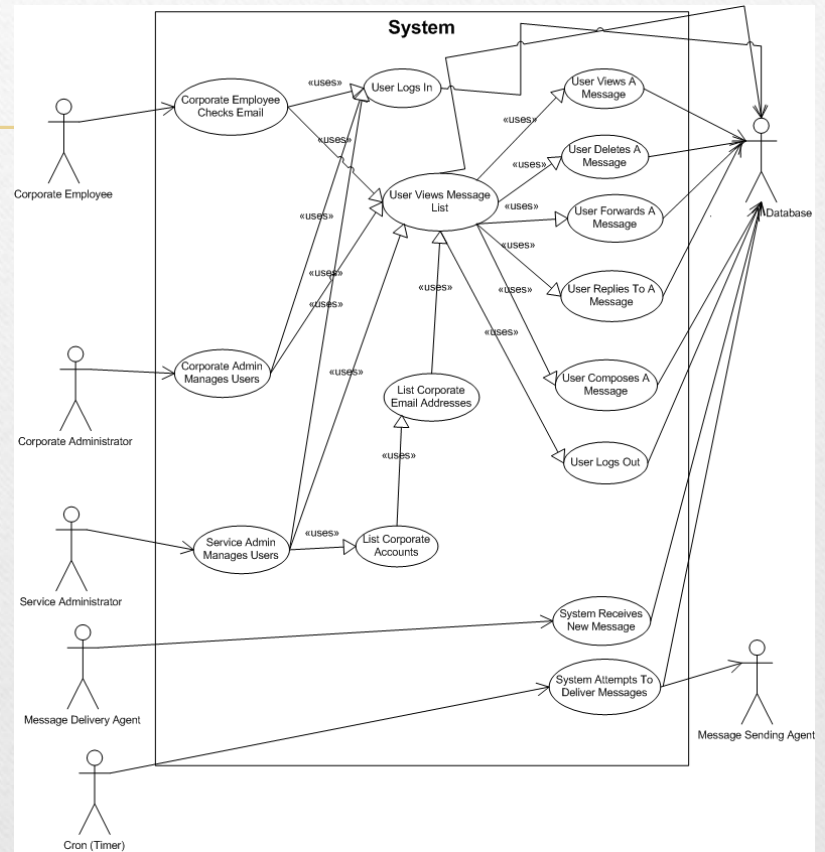
- mire használják a rendszert ?
- mit csináljon a rendszer ?
- hogyan használják a rendszert ?
- mit tudjon a rendszer ?

USE CASE bonyolultsága

Kezdeti:

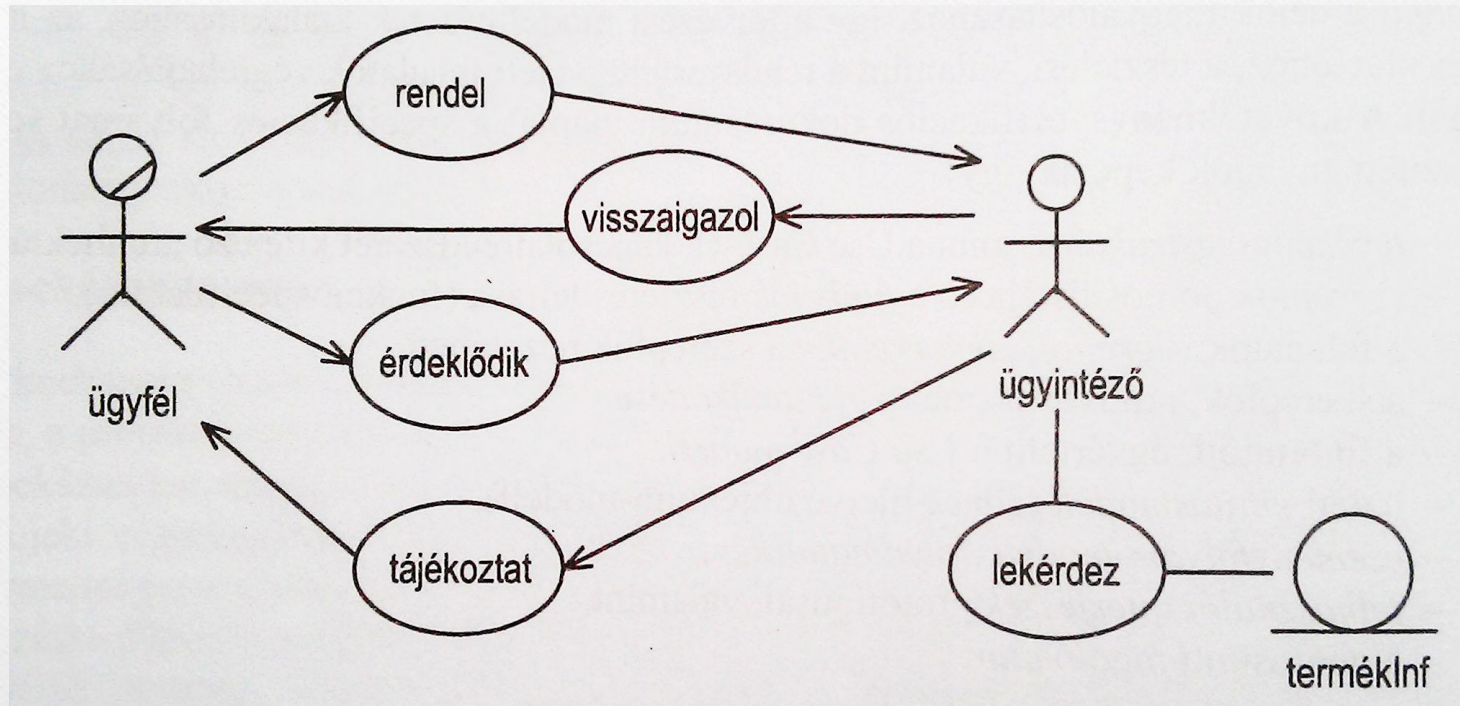


Végleges/átgondolt:



USE CASE technika

Kommunikáció (relációhalmaz) lehet egyirányú, vagy kölcsönös (nyilakkal jelölhetjük, ha hangsúlyozni akarjuk az üzenet irányát)



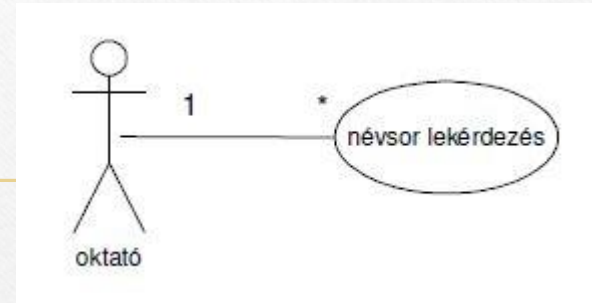
Use Case-nek **teljesnek** kell lennie (tovább nem bontható), és akkor befejezett, ha **előállította a végterméket**, és azt **továbbította** is a címzettnek.

USE CASE Relációk

Relációk: a kapcsolatot jelentik

Társítás (asszociáció):

- jele folytonos vonal, aktor és használati eset között

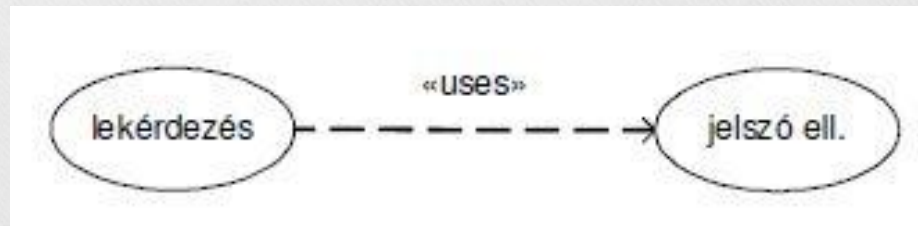


Általánosítás (generalization):



Include: <<include>> (1.5 UML szabvány szerint) vagy <<uses>>

Két használati eset között áll fent, ha az egyik magában foglalja a másikat. **Az egyik használati eset használja, és mindig használja a másikat**



USE CASE Relációk

Kiterjesztés: <<extend>>

kibővítés (kivételkezelés, hibakezelés)

az egyik használati eset működését kiegészíti egy másik használati eset.

Pl.: jelszó ellenőrzés, hibás jelszó kezelése

A „jelszóellenőrzés” **használati esetet** kibővíthetjük egy olyan funkcióval, amely lekezeli azt, ha a felhasználó hibás jelszót ad meg.

Fontos résztvevőkenységek leválasztása



Objektummodell megtervezés

- Osztályok specifikálása → osztály-sajátosságok, asszociációk, öröklődési hierarchia és aggregáció és ezek pontos jellemzése
- Attribútum- és műveletszignatúra
- Osztályok felelősségei, szolgáltatásai (CRC-kártyák, Interfész-tervek)
- Felhasználói interfészek pontos terve (képernyőképek)
- Esemény- és folyamatmodell készítése
- A rendszer viselkedése (állapotok modellezése, a specifikációk pontosítása)

A kifejlesztett alkalmazás implementálása, tesztelés és telepítés

Rövid szöveges leírás, egy *elképzelt* implementálásról,
telepítésről..

Esetleg oktatás megszervezésről, stb...

A fejlesztés implementálása WinDev-ben és ennek dokumentálása

Képernyőképek, és annak
szöveges értelmezései
elegendőek....

Konkrét fejlesztést nem várok el!!!

Féléves feladathoz kiadott mérföldkövek:



SZÉCHENYI
ISTVÁN
EGYETEM



CASE-technológia [NGB_IN027_1]

Összegző dokumentáció a

című feladat fejlesztésének megvalósításáról

Témát kidolgozó
csoport kódja: --

Készítették:

_____[]
_____[]
_____[]

Győr, 2015. _____. ____.

A fejlesztési dokumentum tartalma, fejezetei (aláhúzott rész kötelezően elkészítendő):

- 1) Bevezetés
 - témaválasztás indoklás, cél pontosítása
- 2) Problémater-elemzés
 - probléma pontos definiálása
 - (régli) és új rendszer előnye, hátránya
 - SWOT-analízis
 - Ishikawa elemzési módszer
 - Pareto-analízis
 - Szervezeti struktúra
 - Folyamatgráf
- 3) Megoldási javaslat(ok) a problémára
 - megoldási javaslatok felvetése (minimum 2-3.)
- 4) Megoldási javaslat(ok) kiértékelése, döntés a megoldásváltozatról
 - megoldási javaslatok kiértékelése (táblázatosan, pontozásos módszerrel.)
- 5) Követelmények pontos specifikációja (<http://creately.com>)
 - USE CASE diagramok (<http://creately.com/diagram-type/use-case>)
 - Folyamatábrák (min 4. folyamat)
 - Folyamatgráf
 - Forgatókönyvek
- 6) Objektummodell megtervezése (pontos specifikációkkal):
 - Interfész-tervek, ábrótváz tervek (mockup, wireframe)
[<http://pencil.evolus.vn/> vagy <http://balsamiq.com/>]
 - Több javaslat [<http://www.badlanguage.net/seven-website-mockup-tools>]
 - Osztályok specifikálása → osztály-sajátosságok, asszociációk, öröklődési hierarchia és aggregáció és ezek pontos jellemzése
 - Attritúum- és műveletszignatúra
 - Osztályok felelősségei, szolgáltatásai (CRC-kártyák)
 - Esemény- és folyamatmodell készítése
 - A rendszer viselkedése (állapotok modellezése, a specifikációk pontosítása)
- 7) A fizikai leképezés feladatai:
 - Pontos adatbázis specifikáció (<http://www.mysql.com/products/workbench/>)
 - A működtetendő rendszer hardver- szoftver környezetének meghatározása (különös hangsúlyt helyezve az adattárokra és az adatbázis-kezelőkre)
 - Programspecifikáció elkészítése az 5. pontban definiáltak alapján (architektúra- és programterv)
 - Tesztelési terv, tesztesetek, tesztelési kritériumok, a végrehajtás folyamatának terve
 - Telepítési terv (deployment)
- 8) A kifejlesztett alkalmazás implementálása (WinDev-ben), tesztelés, telepítés és ennek dokumentálása
 - Képernyőképek elkészítése egy választott programban

WinDev példa

Mintafeladat letölthető:

<http://rs1.sze.hu/~takacsg>

Oktatás → **CASE-technológia (GB_IN027_1)**

Letölteni: wd_ex.zip (949MB → 1,1GB)

Szabadidőközpont példa + Winder 15 express