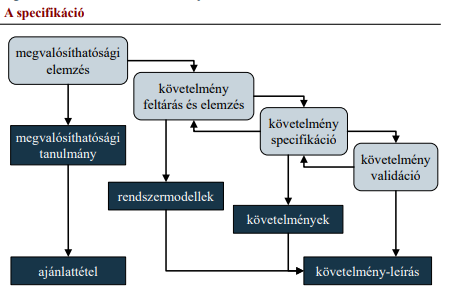
# Követelmények meghatározása



* Célja a feladatot megoldó szoftver funkcióinak tisztázása, a rendszerre és a fejlesztésre vonatkozó elvárások megadása.
* Lépései:
  1. megvalósíthatósági elemzés
  2. követelmény feltárás és elemzés
  3. követelmény specifikáció
  4. követelmény validáció
* Eredménye: a szoftver követelmény-leírása.

### 1. Megvalósíthatósági elemzés

* Azt vizsgálja, hogy a szoftver megvalósítható-e, és üzletileg megtérül-e a befektetés. (erőforrások, költség, időtartam, üzemeltetési feladatok/költségek).
* Az elemzést gyorsan, alacsony költségekkel kell elvégezni.
* Az elemzést ajánlattétel követi a vevő számára.

### 2. Követelményfeltárás és elemzés

Interaktív folyamat, lépései:

1. kutatás: meglévő rendszerek tanulmányozása,   
   egyeztetés, interjúk a lehetséges felhasználókkal
2. osztályozás: a követelmények rendezése, csoportosítása
3. prioritások meghatározása: fontossági sorrend felállítása, ütközések feloldása, szükségtelen követelmények elhagyása

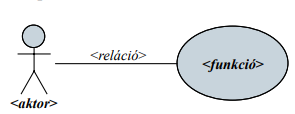
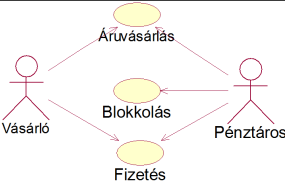
A követelmények alapján létrehozhatók a rendszer modelljei, prototípusai (pl. felhasználói felület terve, menürendszer).

A követelmények típusai:

* **funkcionális**: a szoftver által nyújtott szolgáltatások, reakciók, viselkedések leírása (mit csinál a program?)
* **nem funkcionális**:
  + minőségi követelmények:
    - hatékonyság: erőforrások takarékos használata, sebesség,
    - megbízhatóság: funkcionalitás garantálása (pl. adatbevitel ellenőrzése),
    - biztonság: tárolt adatok védelme (pl. bejelentkezés jelszóval),
    - hordozhatóság: milyen platformokon használható, hogyan vihető át, (pl. Windows 7+, OS/X, Linux)
    - felhasználhatóság: könnyű tanulás, telepítés, üzemeltetés
  + menedzselési követelmények:
    - környezeti: igényelt erőforrások (pl. 4 GB RAM, X GB HDD), szoftverek (pl. Windows 7+, JRE, MySQL szerver)
    - működési: felhasználás paraméterei (mikor, hányan használják), igényelt szaktudás
    - fejlesztési: fejlesztőkörnyezet, nyelvek, könyvtárak (pl. Java, MySQL)
  + külső követelmények:
    - jogszabályoknak való megfelelés
    - etikai kérdések (felhasználók adataival)
    - tulajdonjogi kérdések

### 3. Követelmény specifikáció

Az elemzés során kapott információk egységes alakba foglalva. Történhet beszélt nyelven vagy formalizált leírással (pl. UML use case diagram).

A funkciók részletesebb kifejtését teszik lehetővé a felhasználói történetek (user story), amelyeket agilis fejlesztéseknél használnak.

* a felhasználók által végzett tevékenységsorozatokat és azok lehetséges kimeneteleit adják meg különböző szituációkban
* a történetekben azonosítunk egy szerepkört és egy célt, valamint megindokoljuk a cél szükségességét  
  Például: Mint tanár szeretném megtartani ezt az órát, hogy a tanulók megtanulják a tananyagot.
  + Az indoklás elhagyható, ha egyértelmű.
  + A történetet rögzített kulcsszavakkal írjuk le. (As a … I want to … so that …)
* Az egyes történetekhez megadják az elfogadási feltételeket (acceptance criteria), amelyek egyúttal megadják a végleges alkalmazás funkcinális teszteseteit is.

## 4. Követelmény validáció

Ellenőrzi a megadott specifikációt, és ennek alapján tesztesetek készülhetnek.

## Követelmény-leírás

A specifikáció eredménye. Tartalmazza a szoftver:

* célját, szükségességét, előnyeit, előzményeit
* a funkcionális és nem funkcionális követelmények leírását felhasználói és fejlesztői szemszögből
* a rendszer felépítését és modelljeit
* javaslatokat, becsléseket a megvalósításhoz, a továbbfejlesztéshez és a karbantartáshoz
* a technikai kifejezések leírását

## Források:

* <http://people.inf.elte.hu/groberto/elte_szt/eloadas_anyagok/elte_szt_ea01_dia.pdf>
* <http://people.inf.elte.hu/groberto/elte_szt/eloadas_anyagok/elte_szt_ea02_dia.pdf>