UML (Unified Modeling Language)

# UML nélkül

* **Kommunikációs szakadék van a megrendelő és a fejlesztők között.**
  + Agilis módszereket be kell vonni.
  + Prototípusokat kell fejleszteni és azokat véleményeztetni.
  + Meg kell találni a közös nyelvet.
* **Kommunikációs szakadék van fejlesztő és fejlesztő között**
  + Tapasztalat, tudásszint béli különbségek lehetnek.

# Alapelvek

* Grafikus leírónyelv, ami segít vizualizálni, specifikálni, tervezni és dokumentálni.
* **Kinek jó?**
  + Megrendelő egy folyamatábrát könnyen tud értelmezni.
  + Fejlesztő könnyebben megérti, hogy a másik fejlesztő rendszere hogy működik.
  + Vizuális ábrázolás jobb megértést biztosít.
  + Dokumentáció és így alapos lesz általa.

## UML használata

* UML egy szigorú modellező nyelv.
* **Modellező eszközök:**
  + Microsoft Visio
  + Visual Paradigm for UML
  + Rational Rose

# Célok

* Egyszerűsíti a bonyolult struktúrákat.
* Kommunikációs eszközként szolgál.
* Automatizálja a szoftverek előállítását és folyamatokat.
* Segíti a szerkezeti problémák megoldását.
* Javítja a munka minőségét.

# Diagramok bemutatása (kategóriák: Deployment / Behavioral / Structural / Implementation).

<https://www.visual-paradigm.com/guide/uml-unified-modeling-language/behavior-vs-structural-diagram/>

<https://www.tutorialspoint.com/uml/uml_deployment_diagram.htm>

https://people.inf.elte.hu/treszka/oktatas/OOP/11.pdf

## Structural

### Structural – Composit Structure Diagram

### Structural – Deployment Diagram

### Structural – Package Diagram

### Structural – Profile Diagram

### Structural - Class Diagram

### Structure – Object Diagram

### Structure – Component Diagram

## Behavioral

### Behavioral – Use Case Diagram

* **Célja:** Megrendelővel való egyeztetés, hogy pontosan milyen szerepköröket, funkciókat képzelt el, és ezeket a funkciókat melyik szerepkörrel lehet igénybe venni.
* **Aktor és használati eset közötti megfeleltetés.**
* **Van öröklődés:**
  + Aktorok között
  + Használati esetek között
* **Tartalmazás/Kibővítés**
  + Használati esetek között

### Behavioral – Activity Diagram

* Use-case utáni következő lépés.
* **Célja**:
  + Rendszer folyamat lerajzolása.
  + Leírja az egyik tevékenységtől a másikig tartó sorrendet.
  + Leírja a rendszer párhuzamos, elágazó és egyidejű folyamatát.
* A lépésenkénti tevékenységek és műveletek munkafolyamatainak grafikus ábrázolása.
* **Használati eseteknél a cél:**
  + Belső folyamatok ábrázolása
  + Egymás után következőségek ábrázolása
* 1 használati eset = 1 Activity diagram
* Egy diagrammon belül más use case-ek is előjöhetnek.

### Behavioral – State Machine Diagram

* Rendszer/Objektum állapotainak egymás után következőségét ábrázolja.
* Irányított gráf, aminek csomópontjai a logikai állapotok, amik élei a köztük lévő átmenetek.
* A végrehajtható műveletek az állapotokhoz és az átmenetekhez is tartozhatnak.

### Behavioral – Interaction Diagram

### Behavioral – Sequence Diagram

### Behavioral – Communication Diagram

### Behavioral – Interaction Overview Diagram

### Behavioral – Timing Diagram

## Deployment