

# Szoftver management

## A fejlesztési folyamat részei

### Programkészítés lépései

#### 1. Specifikáció

- a. A megrendelő kitalálja, hogy milyen programot szeretne.
- b. Megbeszéli a programozóval, hogy mit kell tudnia a programnak.
- c. **A tisztázott igények precíz leírását nevezzük specifikációnak.**

#### 2. Tervezés

- a. A feladathoz megfelelő adatszerkezeteket és algoritmusokat kell találni vagy fejleszteni.
- b. Megtervezendő továbbá a program felhasználói felülete és gondolni kell a jövőbeli bővíthetőségre is.

#### 3. Kódolás

- a. A kész terveket egy választott programozási nyelven kódoljuk.
- b. A forráskódból egy fordítóprogram hozza létre a futtatható programot.

#### 4. Tesztelés

- a. Két szempont alapján kell tesztelni a programot:
  - i. **Minden esetben működik-e és hatékony-e.**

#### 5. Hibajavítás

- a. A felismert hibák kijavításához új specifikáció, új terv és újrakódolás lehet szükséges.

#### 6. Dokumentáció

- a. Ha később is szeretnénk fejleszteni a programot és érteni akarjuk a működését, akkor egy dokumentációt kell készíteni hozzá.
- b. Dokumentációnak **két szintje** van:
  - i. **Felhasználói és Fejlesztői**

### Szoftvertervezés egy lehetséges menete

#### 1. Deployment diagrammok

- a. A rendszer és milyen körülmények között lesz használva.

#### 2. Behavioral diagrammok

- a. Milyen funkciókat kell a rendszernek tudnia?
- b. Use-case + Activity + Wireframes diagrammok
- c. Megrendelővel közös tervezés

#### 3. Structural diagrammok

- a. A működést milyen modulokkal, milyen felbontással lehet megoldani.
- b. Component + Sequence (Class diagrammok)
- c. Entity-relations diagram (Adatbázis tábla struktúrák)

#### 4. Időtervezés

- a. Gantt diagram

#### 5. Implementálás

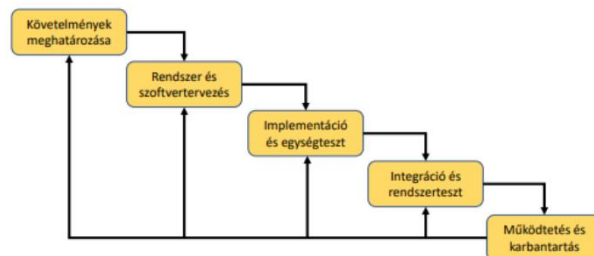
## Fejlesztési folyamat közbeni problémák

- **Követelmények változása:** Ügyfelek igényei fejlesztéskor folyamatosan változhatnak.
- **Híányos tesztelés:** Hibás alkalmazásokat vagy váratlan problémákat okozhatnak.
- **Hibás kommunikáció**
  - o Fejlesztők és az ügyfelek között, de lehet a fejlesztői csapaton belül is rossz kommunikáció.
- **Technikai korlátok**
  - o Akadályozhatja a fejlesztés menetét, minőségét az, hogy nem állnak rendelkezésre a megfelelő eszközök, technológiák a fejlesztés során.

## Klasszikus modellek

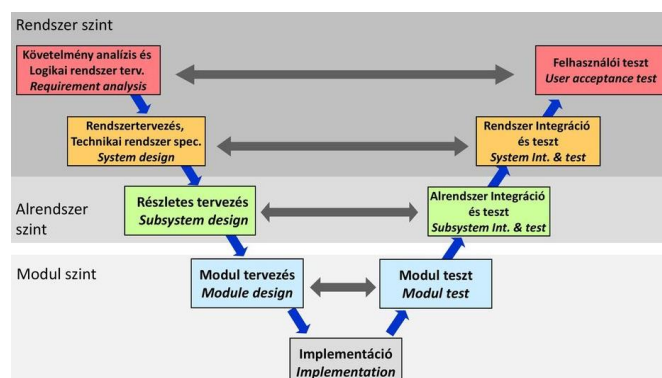
### Vízesés modell

- Akkor hasznos, ha a követelmények jól ismertek és csak nagyon kis változások lehetségesek a fejlesztéskor.
  - o Kevés üzleti rendszernek vannak stabil követelményei.
  - o Főleg nagy rendszerek fejlesztésekor használják, ahol a fejlesztés több helyszínen történik.
- **Problémái**
  - o Minden a specifikáció minőségétől függ.
  - o Későn lát a megrendelő működő programot.
  - o Kezdeti bizonytalanságot nehezen kezeli.
  - o Tesztelés szerepe nem eléggé hangsúlyos.
- **Fázisai**



### V-modell

- Azért nevezik V-modellnek, mert két szára van: **Fejlesztési** és **tesztelési** szár.
- Vízesés modell kiegészítése teszteléssel.
  - o Először végre kell hajtani a fejlesztés lépéseit, ezután jönnek a tesztelés lépései.
  - o Ha valamelyik teszt hibát talál, akkor vissza kell menni a megfelelő fejlesztési lépésre.
- Szigorú dokumentálást követel és nem küszöböli ki a vízesés modell problémáit.



# Agilis módszerek

## Scrum

- **Vezetőség az erőforrásokat adja:** Emberi, anyagi és infrastrukturális
- Adott Scrum Team közvetlenül kommunikál a megrendelővel.
- Teljesen az ő felelősségük a projekt esetleges kudarca.

## Scrum szerepkörök

- **Product Owner**
  - o Csapat tagja, megrendelő érdekeit képviseli.
  - o Megszervezi a rendszeres demókat.
  - o Nem fejlesztő, de érti a gazdasági folyamatokat.
  - o Product Backlog kezelője.
- **Scrum Master**
  - o Csapat tagja, egy hagyományos projektmenedzser szereppel egyezik a feladata.
  - o Felügyeli a folyamatokat.
  - o Konfliktusokat kezel, akadályok elhárítását irányítja.
  - o Meetingeket ő szervezi, ő vezeti.
  - o Scrum csapat feje.
- **Scrum Team**
  - o 5-9 fő alkotja, **szükséges:** Elemző, fejlesztő, tesztelő
  - o Ők végzik a tényleges fejlesztést.
  - o Felelősségük, hogy egy sprintre bevállalt feladatokat elvégezzék.
  - o **Fejlesztői fokozatok:** Junior > Medior > Senior
  - o **Tesztelő**
    - Mindenki a saját kódját teszteli.
    - Unit tesztek után peer review más fejlesztővel.
    - Van manuális teszt, amit nem a fejlesztők végeznek.

## Scrum entitások

- **User Story, Task**
  - o Specifikációból eredő feladat.
  - o Taskokra bomlik szét, ezeket veszik magukra a fejlesztők.
- **Sprint**
  - o 1-4 hét hosszú fejlesztési szakasz.
  - o Addig jönnek újabb sprintek, amíg a Product Backlogból el nem tűnnek a User Storyk.
  - o A sprint vége egy kész szoftver.
- **Product Backlog**
  - o Elkészítésre váró Story-k gyűjtőhelye.
  - o Product Owner tartja karban, tehát prioritásokat rendel a story-khoz.
  - o **ROI:** Return of Investment = üzleti érték / ráfordítás.
- **Sprint Backlog**
  - o A Product Backlog, de csak az adott sprintre bevállalt storykra szűrve.
- **Burn down/up chart**
  - o Napi eredmény diagram és megmutatja, hogy a csapat mennyire tartja az eredeti ütemtervet.
- **Impediment**
  - o Akadály, ami a munkát hátráltatja, valamilyen munkahelyi probléma.
  - o Scrum Master feladata eljárítani.

### *Indító meetingek*

- **Sprint (pre)grooming**
  - o Architektek + Scrum Master
  - o Storyk ellenőrzése, pontozása (nehézség/idő > Fibonacci számokkal)
- **Sprint planning**
  - o Team bevállal storykat.
  - o Figyelembe veszik a pontozást.

### *Folyamatos meetingek*

- **Daily stand-up**
  - o Minden nap ugyanakkor és az egész team részt vesz benne.
  - o Szóban, kb 15 perc és egyéb eszközök nélkül.
  - o Megbeszéljük, hogy ki mit csinált vagy ma mit fog csinálni, és kinek mi a problémája.
- **Sprint refinement**
  - o Hetente 1 maximum (3 hetes sprintben maximum 2 alkalom)
  - o Storyk/Taskok áttekintése
  - o Gyors hibaelhárítások
  - o Nem része a Scrumnak, de alkalmazzák.
- **Egyéb meeting**
  - o Csapat részhalmaza egyeztet.

### *Lezáró meetingek*

- **Sprint review**
  - o Team + Scrum Master + Product Owner
  - o Eredmények bemutatása, Product Owner dönti el, hogy sikeres-e a sprint.
  - o Kimaradt Storyk/Taskok a következő sprintre mennek.
- **Sprint retrospect**
  - o Review után tartják, Team + Scrum Master
  - o Személyes tapasztalatok/javaslatok megvitatása.
  - o Céges szintű problémákat a Scrum Master továbbítja felfelé.
  - o Személyes konfliktusok megbeszélése.

### *Kanban (jelzőtábla vagy hirdetőtábla).*

- Lean módszer emberek csoportos munkájának menedzselésére és fejlesztésére.
  - o **Lean:** Vállalatirányítási módszer, aminek célja, hogy a vállalat minél gazdaságosabban állítsa elő a termékeit, szolgáltatásait.
- Kiegyensúlyozza az igényeket és elérhető munkakapacitást.
- Munkafolyamat megjelenítésére használják a Kanban táblát.
- A munkafolyamatok elvégzését úgy ütemezik, ahogy a dolgozók kapacitása megengedi és nem engedik, hogy a munkafolyamatok sürgőssége határozza meg a munka időbeosztását.

### *Kanban célja*

- Célja, hogy létrejöjjön egy olyan vizuális folyamatmenedzsment-rendszer, ami segít meghozni azokat a döntéseket, hogy mit, mikor és hogyan gyártsanak.

### *Hol használják?*

- Maintenance/support jellegű feladatoknál
- HR-folyamatokra
- Értékesítésre