

Szoftver menedzsment

• Programkészítés lépései

1. Specifikáció → Fejlesztő ügyféllel megbeszéli, hogy milyen funkciókat szeretne, ezt nevezünk pontos specifikációnak.
2. Tervezés → Követelmények, diagramok elkészítése
3. Kódolás → Fejlesztő dönti el, hogy milyen nyelven fejleszti le a szoftvert.
4. Tesztelés → Legyen minden **zöld**
5. Hibajavítás → Kivételkezelés
6. Dokumentáció → Felhasználói és fejlesztői

• Konkrét lépések

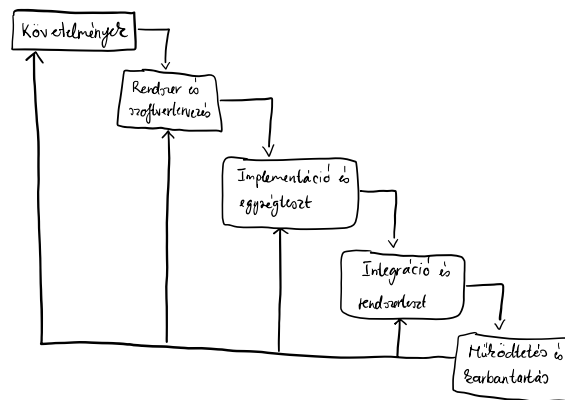
1. Deployment diagram → Rendszer követelmények
2. Behavioral diagram → Activity, Use-case diagram, stb...
3. Structural diagram → Package, Component diagram, stb...
4. Időtervezés → Gantt-diagram
5. Implementálás

• Fejlesztési folyamat közbeni problémák

- Követelmények változása → ügyfél meggondolja magát
- Hiányos tesztelés → Hibákkal teli szoftver
- Kommunikáció → Fejlesztő és közölt vagy fejlesztő és ügyfél között
- Technikai korlátok → Régi számítógép, lejárt licenz

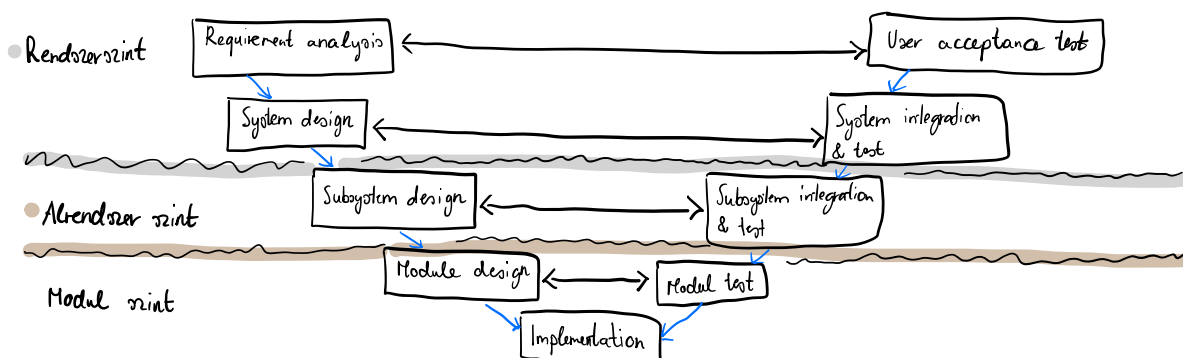
• Vízvezető modell

- Hasznos, ha a követelmények ismertetése és az az változtatások lehetőségei fejlesztőkor
- Problémái:
 - Minden a specifikáció minőségétől függ.
 - Későn lát a megrendelő kész programot.
 - Tesztelés szerepe nem elég hangsúlyos
- Fázisai:



• V-modell

- Két sávra van: fejlesztési és tesztelési sáv
- Először fejlesztés, majd tesztelés és ha hiba van, akkor vissza a fejlesztésre, majd tesztelés.
- Szigorú dokumentálást követel és nem küszöböli ki a vízvezető modell problémáit.



• Scrum

- Agilis módszer
- Vezetőség az erőforrásokat adja → Emberi, anyagi és infrastrukturális
- Adott Scrum Team közvetlenül kommunikál a megrendelővel
 - Teljesen az ő felelősségük a projekt sikerése

• Scrum szereplők

- Product Owner
 - Megrendelő érdekeit képviseli
 - Megszervezi a rendszeres demókat
 - Nem fejlesztő, de érti a gazdasági folyamatokat
 - Product Backlog kezelője
- Scrum master
 - Projektmenedzser
 - Felügyeli a folyamatokat
 - Konfliktusokat kezel
 - Meetingeket ő szervezi és vezeti
 - Scrum csapat fiye
- Scrum Team
 - 5-9 fő, működés: Elemző, fejlesztő, tesztelő
 - Ők végzik a tényleges fejlesztést
 - Sprintre bevállalt feladatokat elvégzik
 - Fejlesztői fokozatok: junior → medior → senior

• Scrum entitások

- User Story, Task → Task-ra bontják szét, ezeket veszik magukra a fejlesztők
- Sprint → 1-4 hét hosszú fejlesztési szakasz és addig jönnek újabb sprintek, amíg a Product Backlogból el nem tűnik a User Story-k
- Product Backlog → Elérhetőre váró Story-k, amit a Product Owner tart szorban és prioritásokat rendel a story-khoz
- Sprint Backlog → A Product Backlog, de csak az adott sprintre bevett story-ra szűve

• Indító meetingek

- Sprint (pre) grooming → Story-k ellenőrzése, pontosítása
- Sprint planning → Team bevállal story-kat és figyelembe veszik a pontosítást

• Folyamatos meetingek

- Daily standup → Szóban, kb 15 perc, ki/mit csinált, problémák
- Sprint refinement → Hetente 1, story-k áttekintése, gyors hibaelhárítás

• Záró meetingek

- Sprint review → Eredményeket bemutatása és PO dönti el, hogy sikeres-e a sprint
- Sprint retrospect → Review után tartják, tapasztalatok megvitatása

• Kanban

- Lean módszer emberek csoportos munkájának menedzselése és fejlesztése.
- Kiegyensúlyozza az igényeket és elérhető munkakapacitást
- A munkafolyamatot elvégzését úgy őrmezi, ahogy a dolgozó kapacitása megengedi és nem engedi, hogy a munkafolyamat sürgőssége határozza meg a munka időbeosztását.
- Célja, hogy létrejöjjön egy olyan vizuális folyamatmenedzsment-rendszer, ami segít meghozni azokat a döntéseket, hogy mit, mikor és hogyan gyártanak.
- Használják: Értékesítés, HR-folyamatokra