

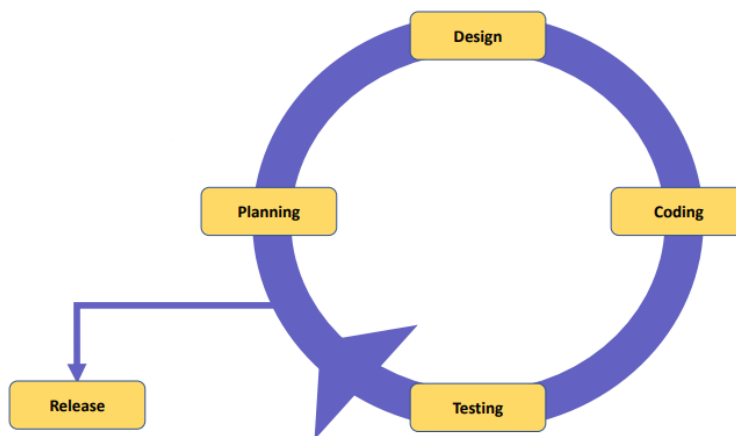
2. Agilis modellek

Agilis módszerek

Extrém programozás

- XP, Extreme programming
- Inkrementális fejlesztés + OOP + Agilis elvek
- 4 aktivitásból áll
 - o **Planning** (analízis)
 - o **Design** (tervezés)
 - o **Coding** (implementáció)
 - o **Testing** (tesztelés)
- **Alapvető értékei**
 - o **Kommunikáció:** Közös gondolkodás, aktív alkotói légkör
 - o **Egyszerűség:** Core funkciók először, apróságok utoljára (core first, shine last)
 - o **Visszacsatolás:** Tesztelők is, megrendelő is, fejlesztők is
 - o **Bátorság:** Merjünk új technológiákhoz nyúlni
 - o **Tisztelet:** Egyén munkája iránt maximálisan

XP modellje



Planning

- User story-k leírása
- Apró release-ek preferálása
- Minden iteráció elején analízis
- Minden nap elején stand-up meeting

Design

- Egyszerűség, CRC kártyák használata, ne hozzunk előre semmit

Coding

- Standard-ek betartása, pair-programming, nincs túlóra

Testing

- Unit-test mindenre, ahol minden teszt legyen zöld a release előtt
- Elfogadási tesztek (acceptance tests) gyakori futtatása

XP előnyök és hátrányok

- **Előnyök**
 - o Jól kezeli a követelmények változását
 - o Költséghatékony
 - o Nagyobb megrendelői elégedettség
 - o Alacsony kockázati tényező
- **Hátrányok**
 - o Minőség erősen függ a megrendelő közreműködő szándékától
 - o Nem terv centralizált, hanem kód-centralizált
 - Nehéz az újrafelhasználása később, nehéz később elővenni
 - o Nem fektet elég hangsúlyt a dokumentáció készítésére

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

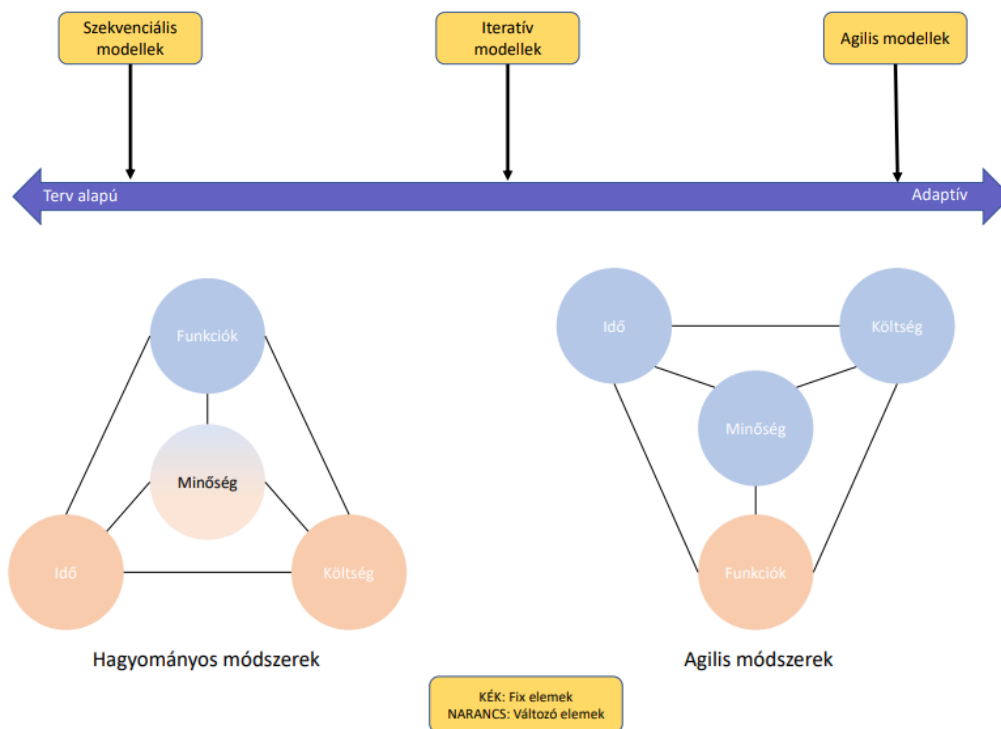
- **Daily stand-up**
 - o Minden nap ugyanakkor és az egész team részt vesz benne.
 - o Szóban, kb 15 perc és egyéb eszközök nélkül.
 - o Megbeszéljük, hogy ki mit csinált vagy ma mit fog csinálni, és kinek mi a problémája.
- **Sprint refinement**
 - o Hetente 1 maximum (3 hetes sprintben maximum 2 alkalom)
 - 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.

Hagyományos modellek és agilis modellek összehasonlítása

- Mindegyik jó a maga módján
- **Szekvenciális modellek**, akkor jók, ha jól át vannak gondolva a dolgok, tehát nem lesznek változtatások és biztosan jól értünk mindent.
 - o **Terv alapú**
- **Agilis modellek**, a megrendelőknek nincsen terve, kb el tudja mondani, hogy mit szeretne, vagyis **Adaptív**, alkalmazkodik a különböző környezeti változásokhoz.
- **Iteratív modellek**, a kettő között helyezkednek el.



- **Hagyományos módszereknél**, a specifikáció fix, idő és költség változó, így a minőség rovására megy.
- **Agilis módszereknél**, a funkciók legyenek változóak, így az időt és költséget jobban lehet rögzíteni, ezzel biztosan kész lesz időre és akkor a minőséget magasabb szintre fogja emelni.

Melyik a sikeresebb?

- Összességében az Agilis módszerek sikeresebben, mivel az elkészült projektek magasabb szinten, időre tudnak elkészülni.
- A hagyományos módszereknél így sokkal nagyobb az esélye, hogy a projekt „elbukik”.