

Szoftvertchnológia

2022 Március 2

3. előadás

A szoftverfolyamat



KADA ZSOLT

STRATÉGIAI ÉS FEJLESZTÉSI
IGAZGATÓ

GIRO ZRT.



PÁZMÁNY PÉTER KATOLIKUS EGYETEM - KIEMELT FELSŐOKTATÁSI INTÉZMÉNY

INFORMÁCIÓS TECHNOLÓGIAI ÉS BIONIKAI KAR - KUTATÓ KAR

Tartalom

1

A SZOFTVERFOLYAMAT

2

A SZOFTVERFOLYAMAT-MODELLEK BEMUTATÁSA

3

A SZOFTVERFOLYAMAT SZAKASZAI



PÁZMÁNY PÉTER KATOLIKUS EGYETEM - KIEMELT FELSŐOKTATÁSI INTÉZMÉNY

INFORMÁCIÓS TECHNOLÓGIAI ÉS BIONIKAI KAR - KUTATÓ KAR

A szoftverfolyamat

Mi is a szoftverfolyamat?

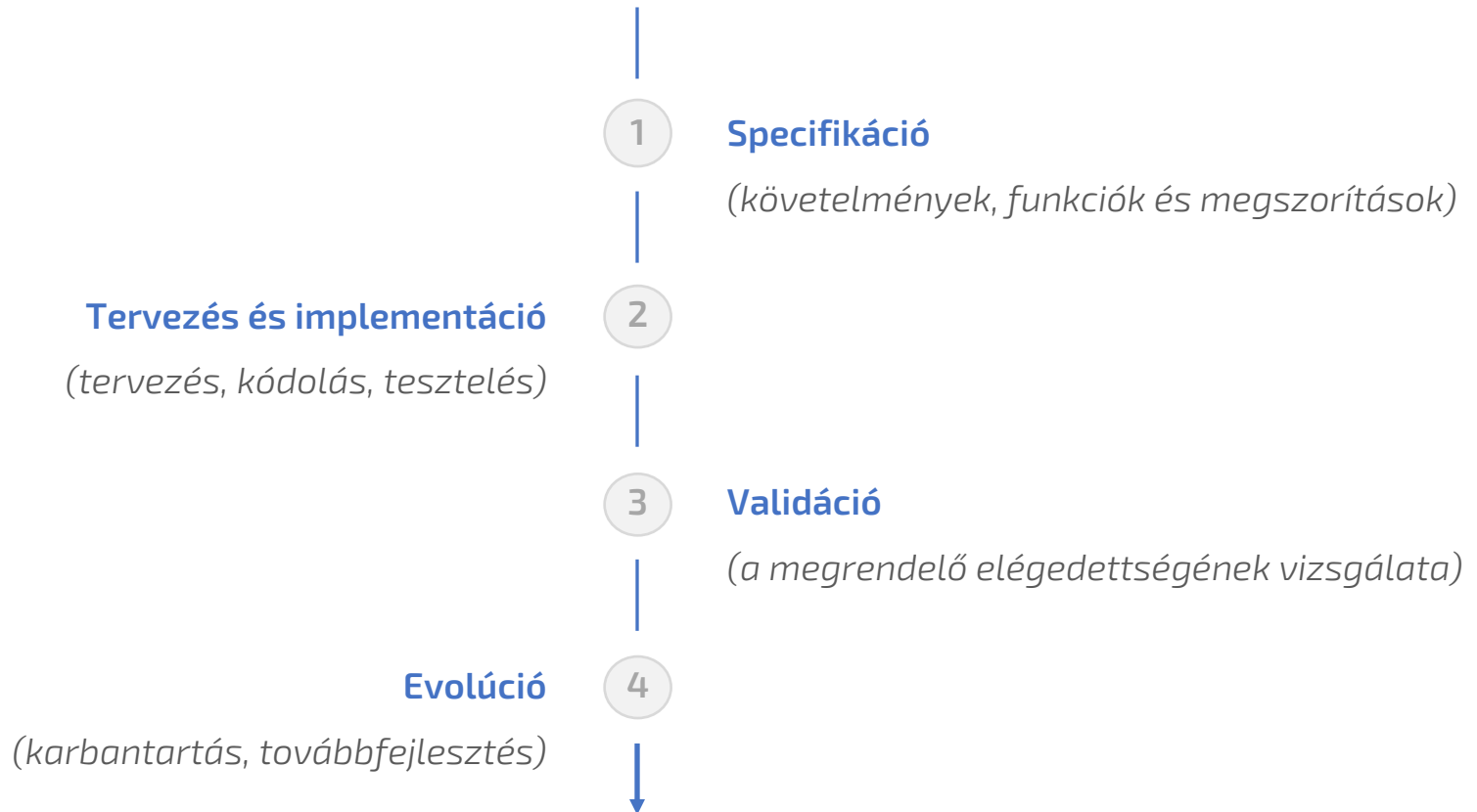
**A szoftverfolyamat egy szoftvertermék előállítására irányuló
tevékenységek sorozata.**

(hozzátartoznak e tevékenységek eredményei is)



A szoftverfolyamat egy szoftver rendszer kifejlesztéséhez szükséges
tevékenységek strukturált halmaza.

FÁZISAI:



Nincs egységes, minden szoftver kidolgozására alkalmas folyamat.
Emberi és szervezeti tényezők fontossága.

Tartalom

1

A SZOFTVERFOLYAMAT

2

A SZOFTVERFOLYAMAT SZAKASZAI

3

A SZOFTVERFOLYAMAT-MODELLEK BEMUTATÁSA



PÁZMÁNY PÉTER KATOLIKUS EGYETEM - KIEMELT FELSŐOKTATÁSI INTÉZMÉNY

INFORMÁCIÓS TECHNOLÓGIAI ÉS BIONIKAI KAR - KUTATÓ KAR

A szoftverspecifikáció



PÁZMÁNY PÉTER KATOLIKUS EGYETEM - KIEMELT FELSŐOKTATÁSI INTÉZMÉNY
INFORMÁCIÓS TECHNOLÓGIAI ÉS BIONIKAI KAR - KUTATÓ KAR

A szoftverspecifikáció

A szoftverspecifikáció – követelménytervezés

- A követelmények felmérése, tervezése a szoftverfolyamat kritikus szakasza, amelynek hibái csak később, a rendszer átadása után derülnek ki.
- A követelménytervezés feladata meghatározni, milyen szolgáltatásokat várnak el a szoftvertől és milyen megkötések érvényesek rá.



A szoftverspecifikáció

A szoftverspecifikáció – követelménytervezés

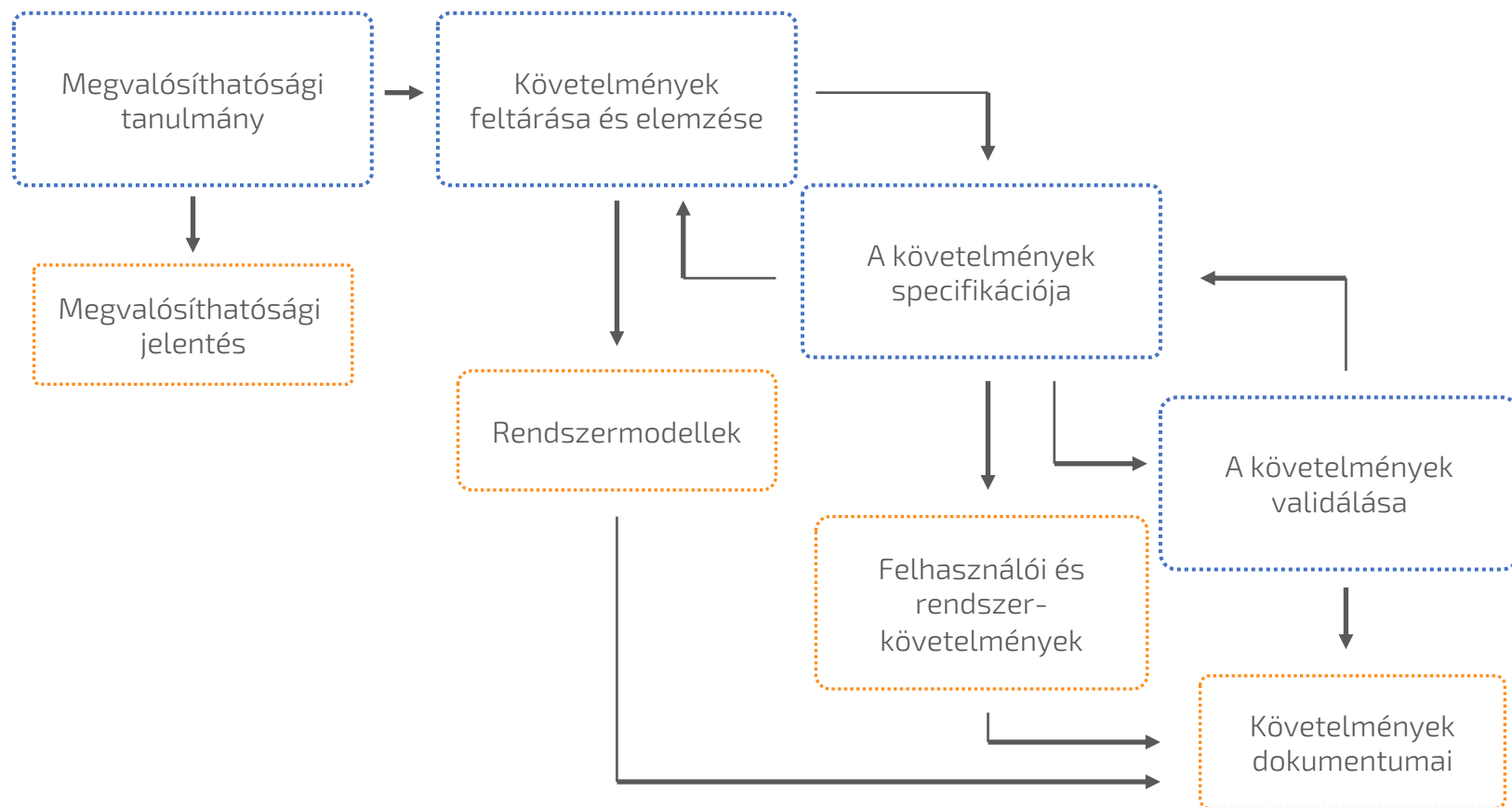
A követelménytervezés folyamata:

1. Megvalósíthatósági tanulmány
(gyakran még a fejlesztést megelőzően készül)
2. Követelmények feltárása és elemzése
3. Követelményspecifikáció
4. A követelmények validálása
(vagyis egyeztetése a megrendelővel)



A szoftverspecifikáció

A követelménytervezés folyamata



Szoftvertervezés és implementáció



PÁZMÁNY PÉTER KATOLIKUS EGYETEM - KIEMELT FELSŐOKTATÁSI INTÉZMÉNY
INFORMÁCIÓS TECHNOLÓGIAI ÉS BIONIKAI KAR - KUTATÓ KAR

Szoftvertervezés és implementáció

- A specifikációk működő rendszerré konvertálásának folyamata.

- **Szoftvertervezés:**

A specifikációkat megvalósító szoftverstruktúra tervezése.

- **Implementáció:**

A struktúra lefordítása futtatható programmá.

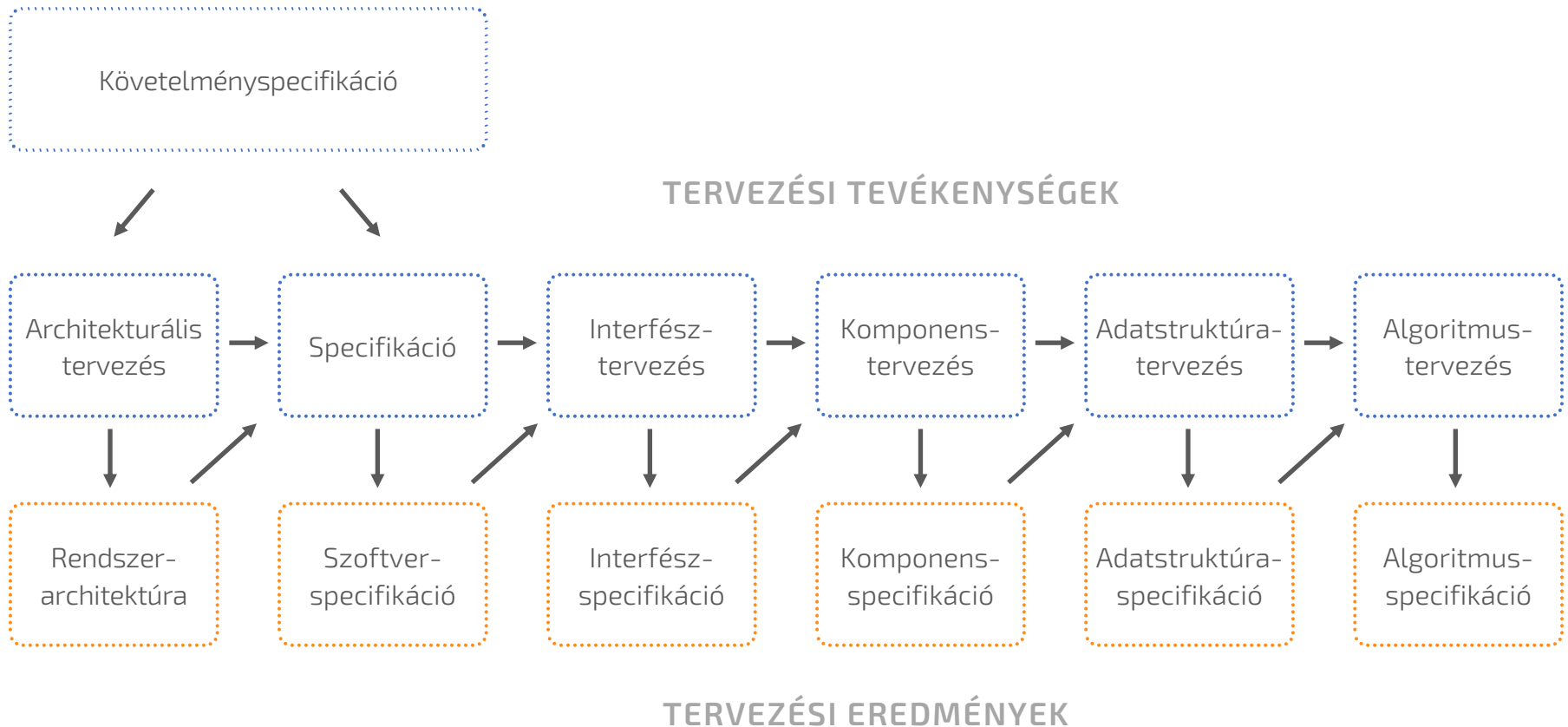
(kódolás, tesztelés)

- A tervezés és az implementáció tevékenységei szorosan összefüggenek és gyakran átlapolhatók.



Szoftvertervezés és implementáció

A szoftvertervezés folyamata



Szoftvertervezés és implementáció

Tervezési módszerek

- Egy szoftverterv kidolgozásának szisztematikus megközelítése.
- A szoftverterveket általában grafikus modellekkel dokumentálják.
- A gyakorlatban például az alábbi modellek használatosak:
 - **Adatfolyam modell**
A rendszert adattranzformációkkal modellezi.
 - **Egyed-kapcsolat modell (ER)**
Az alapvető egyedek és kapcsolataik leírása (*pl. adatbázis*).
 - **Strukturált modell**
A komponensek és kölcsönhatásaik leírása (*általában funkcionális*).
 - **Objektumorientált modellek**
A rendszer használati esetei, osztályhierarchiája, öröklődési modellje, az objektumok statikus és dinamikus kapcsolatai.



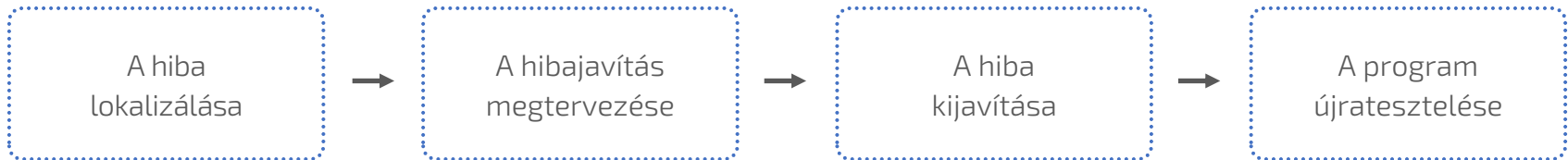
Programozás és nyomonkövetés



PÁZMÁNY PÉTER KATOLIKUS EGYETEM - KIEMELT FELSŐOKTATÁSI INTÉZMÉNY
INFORMÁCIÓS TECHNOLÓGIAI ÉS BIONIKAI KAR - KUTATÓ KAR

Programozás és nyomkövetés

- A terv futtatható programmá transzformálása és a hibák megkeresése, kijavítása.
- A programozás személyes tevékenység, nincsenek általános szabályok, de egy szervezeten belül léteznek konvenciók.
- A programozók tesztelik a kifejlesztett kódot, majd megkeresik a hiba helyét és kijavítják azt (*nyomkövetés, belövés*).
- A belövés folyamata:



Verifikáció és validáció



PÁZMÁNY PÉTER KATOLIKUS EGYETEM - KIEMELT FELSŐOKTATÁSI INTÉZMÉNY
INFORMÁCIÓS TECHNOLÓGIAI ÉS BIONIKAI KAR - KUTATÓ KAR

Verifikáció és validáció

- Célja annak igazolása, hogy a rendszer
 - megfelel a specifikációnak (**verifikáció**) és
 - kielégíti a megrendelő elvárásait (**validáció**)
- Ellenőrzési folyamatok: szemlék, felülvizsgálatok, tesztek.
- A rendszerteszt a rendszer működtetése a specifikációból származó, előre megtervezett tesztesetekkel, amelyek valódi adatokat táplálnak a rendszerbe.



Szoftvertesztelés



PÁZMÁNY PÉTER KATOLIKUS EGYETEM - KIEMELT FELSŐOKTATÁSI INTÉZMÉNY

INFORMÁCIÓS TECHNOLÓGIAI ÉS BIONIKAI KAR - KUTATÓ KAR

Szoftvertesztelés

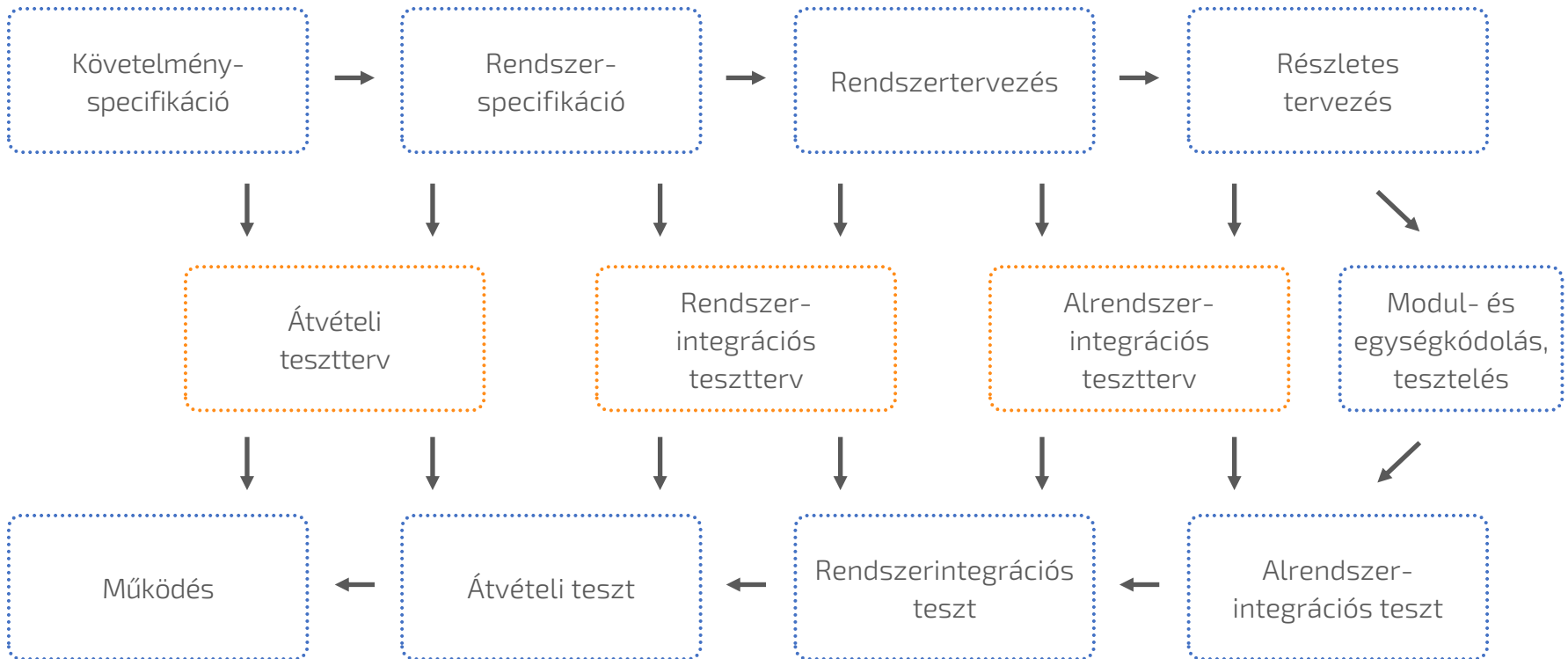
A tesztelési folyamat szakaszai

- **Egységteszt**
 - A komponensek egyedi tesztelése
- **Modulteszt**
 - Egy modul (egymástól függő komponensek) tesztelése
- **Alrendszerteszt**
 - Alrendszerek (több, együttműködő modul) tesztelése
- **Rendszerteszt**
 - A teljes rendszer tesztelése
- **Átvételi teszt**
 - A rendszer tesztelése a megrendelő adataival



Szoftvertesztelés

A tesztelési folyamat szakaszai



Ez tulajdonképpen a V-modell



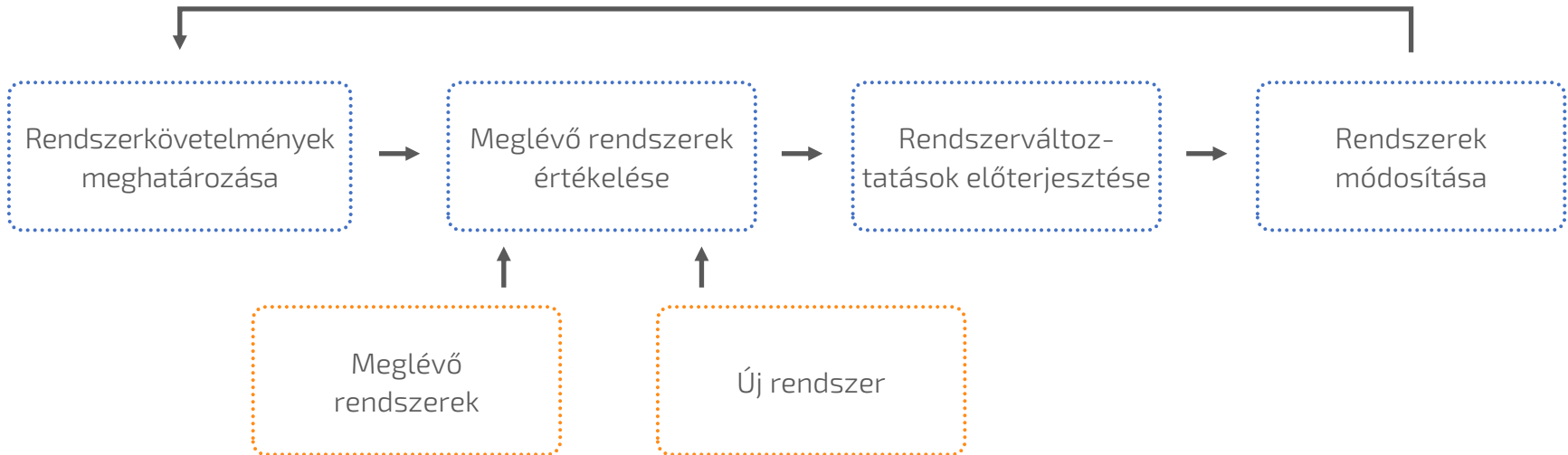
Szoftverevolúció



PÁZMÁNY PÉTER KATOLIKUS EGYETEM - KIEMELT FELSŐOKTATÁSI INTÉZMÉNY
INFORMÁCIÓS TECHNOLÓGIAI ÉS BIONIKAI KAR - KUTATÓ KAR

Szoftverevolúció

- A hardverrel szemben a szoftver könnyebben módosítható (elvileg).
- A változó üzleti követelmények megkövetelik az üzleti folyamatot támogató szoftver megváltoztatását.
- Mind kevesebb a teljesen új rendszer, az újak a régi rendszerekkel összeépülnek, csökken a különbség a fejlesztés és az evolúció (karbantartás) között.



Tartalom

1

A SZOFTVERFOLYAMAT

2

A SZOFTVERFOLYAMAT SZAKASZAI

3

A SZOFTVERFOLYAMAT-MODELLEK BEMUTATÁSA



PÁZMÁNY PÉTER KATOLIKUS EGYETEM - KIEMELT FELSŐOKTATÁSI INTÉZMÉNY

INFORMÁCIÓS TECHNOLÓGIAI ÉS BIONIKAI KAR - KUTATÓ KAR

A szoftverfolyamat-modellek bemutatása

A szoftverfolyamat általános modelljei

- A szoftverfolyamat modellje a folyamat absztrakt reprezentációja. Általános folyamatmodellek: ↓

- **Vízesés modell**

Az alapvető tevékenységek önálló fázisok.

- **Evolúciós fejlesztés**

A specifikáció, a fejlesztés és a validáció összefonódik.

Ez az alapja a manapság elterjedten alkalmazott **agilis** fejlesztési módszereknek.

- **Formális rendszerfejlesztés**

A követelmények matematikai modelljéből formális transzformációval állítja elő a szoftvert.

Ez a módszer az őse a mai modellközpontú fejlesztésnek.

- **Újrafelhasználás-alapú fejlesztés**

A rendszert meglévő komponensek integrálásával állítja elő.



A vízesés modell

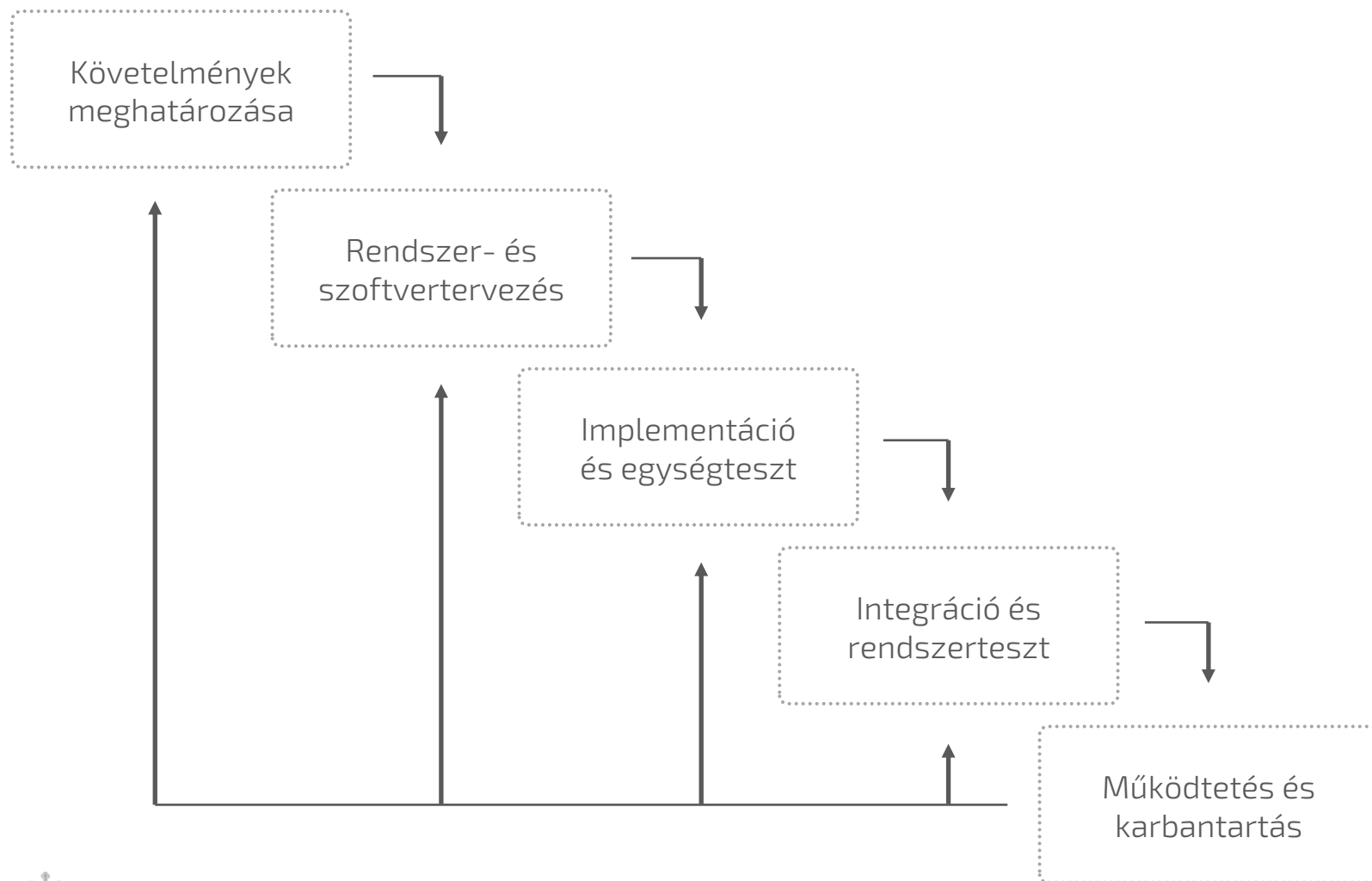


PÁZMÁNY PÉTER KATOLIKUS EGYETEM - KIEMELT FELSŐOKTATÁSI INTÉZMÉNY

INFORMÁCIÓS TECHNOLÓGIAI ÉS BIONIKAI KAR - KUTATÓ



A vízesés modell



A vízesés modell

A vízesés modell fázisai

1. Követelmények elemzése és meghatározása
2. Rendszer- és szoftver tervezés
3. Implementáció és egységteszt
4. Integráció és rendszerteszt
5. Működtetés és karbantartás

Video: → [Waterfall: The Waterfall Model – The Software Experts](#) (10 perc)



A vízesés modell

A vízesés modell előnyei

- ✓ Jól áttekinthető és követhető fejlesztési projektfolyamatot eredményez.
- ✓ A folyamat termékei szerződésekkel könnyen lefedhetők.
(specifikációs és tervezési dokumentumok, program(ok), stb.)
- ✓ A tevékenységek jól és pontosan tervezhetők.



A vízesés modell

A vízesés modell hátrányai

- ✗ Egymástól elkülönült fázisokra osztja a projektet
(*költséges egy korábbi fázishoz visszatérni pl. specifikációs, vagy tervezési hiba esetén*).
- ✗ Csak a projekt végén, az átadáskor (*a működtetés első lépésekor*) derülnek ki a specifikációs hibák.
- ✗ Nem képes rugalmasan alkalmazkodni a felhasználói igények változásaihoz.
- ✗ Csak a követelmények pontos ismeretében alkalmazható.



Az evolúciós fejlesztés



PÁZMÁNY PÉTER KATOLIKUS EGYETEM - KIEMELT FELSŐOKTATÁSI INTÉZMÉNY
INFORMÁCIÓS TECHNOLÓGIAI ÉS BIONIKAI KAR - KUTATÓ KAR



Az evolúciós fejlesztés

- Az alapgondolat: ki kell dolgozni egy kezdeti implementációt, amelyet a felhasználó véleményezhet, és azt kell finomítani az elfogadásig.

- **Feltáró fejlesztés**

A követelmények feltárása lépésenként, a megrendelővel együttműködve történik, folyamatosan kiegészítve a rendszert az új funkciókkal, részekkel.

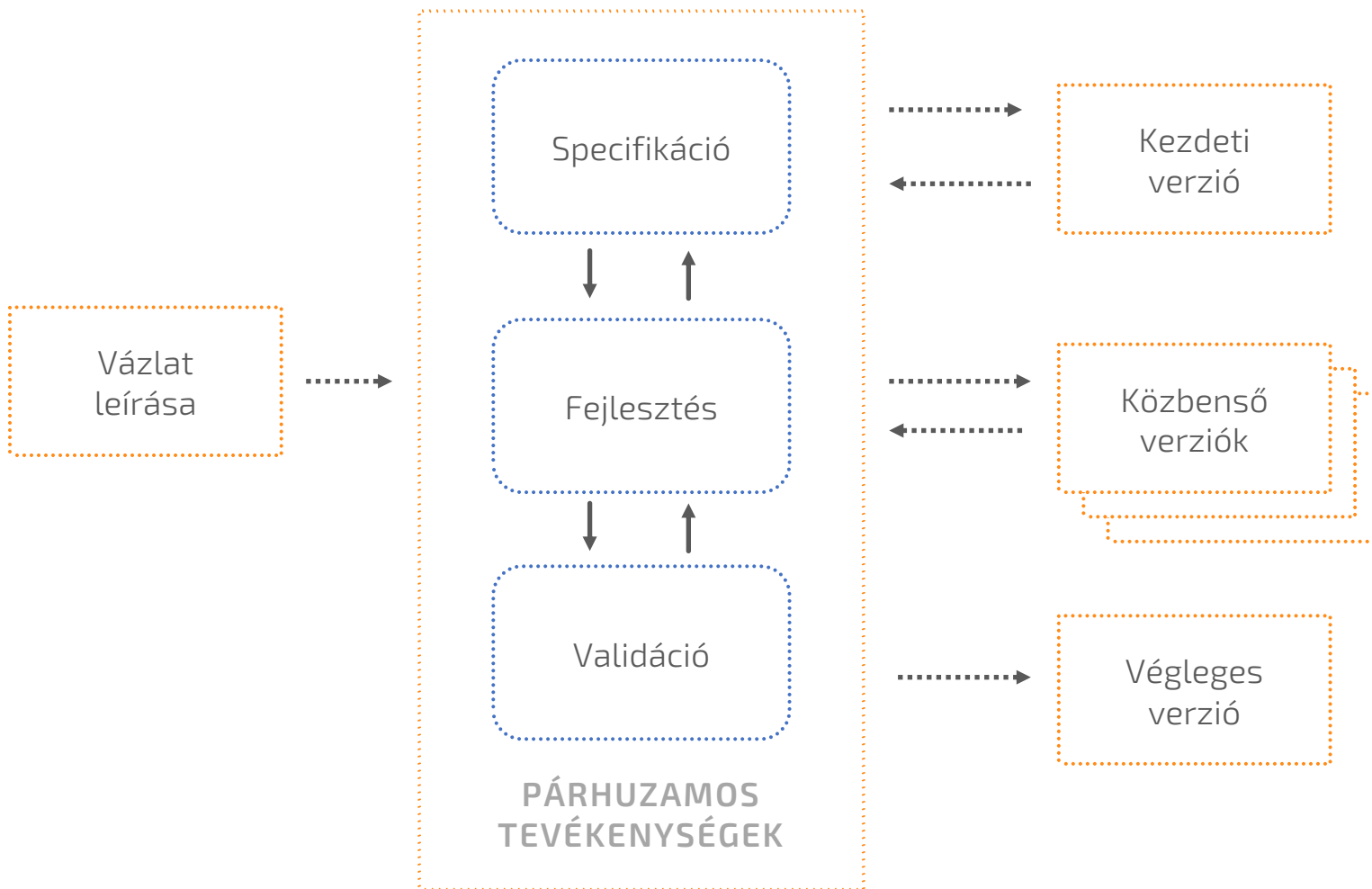
(Ilyen az agilis módszerek többsége.)

- **Eldobható prototípus**

„Deszkamodellek” készítése és átadása az ügyfélnek a követelmények pontosabb feltárása érdekében.



Az evolúciós fejlesztés



Az evolúciós fejlesztés

Az evolúciós fejlesztés előnyei/hátrányai



Előnyök:

- A validáció már fejlesztés közben megkezdődhet.
- Főleg kis- vagy közepes interaktív rendszerek, vagy nagy rendszerek felhasználói interfészének fejlesztésére alkalmas.
- Rövid életciklusú rendszerek esetén előnyös.



Hátrányok:

- A projekt előrehaladása nem követhető.
- A rendszerek struktúrájával nem foglalkozik.
- Nincs részletes specifikáció.
- Speciális eszközöket és ismereteket igényel
(pl. gyors modellezőeszközök alkalmazása:
vizuális programfejlesztés, adatbázis-
programozás /4GL).



Az evolúciós fejlesztési modell az alapja az AGILIS módszereknek.



Formális rendszerfejlesztés

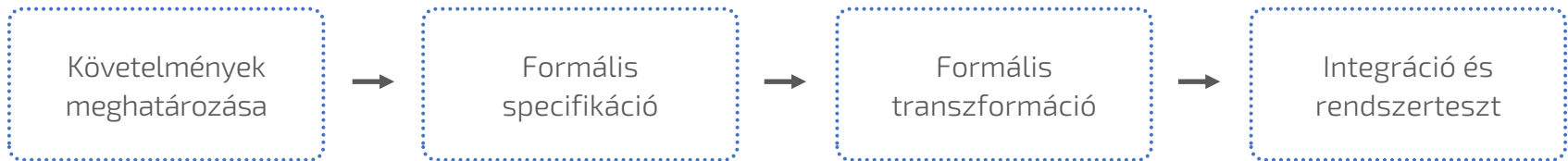


PÁZMÁNY PÉTER KATOLIKUS EGYETEM - KIEMELT FELSŐOKTATÁSI INTÉZMÉNY
INFORMÁCIÓS TECHNOLÓGIAI ÉS BIONIKAI KAR - KUTATÓ KAR



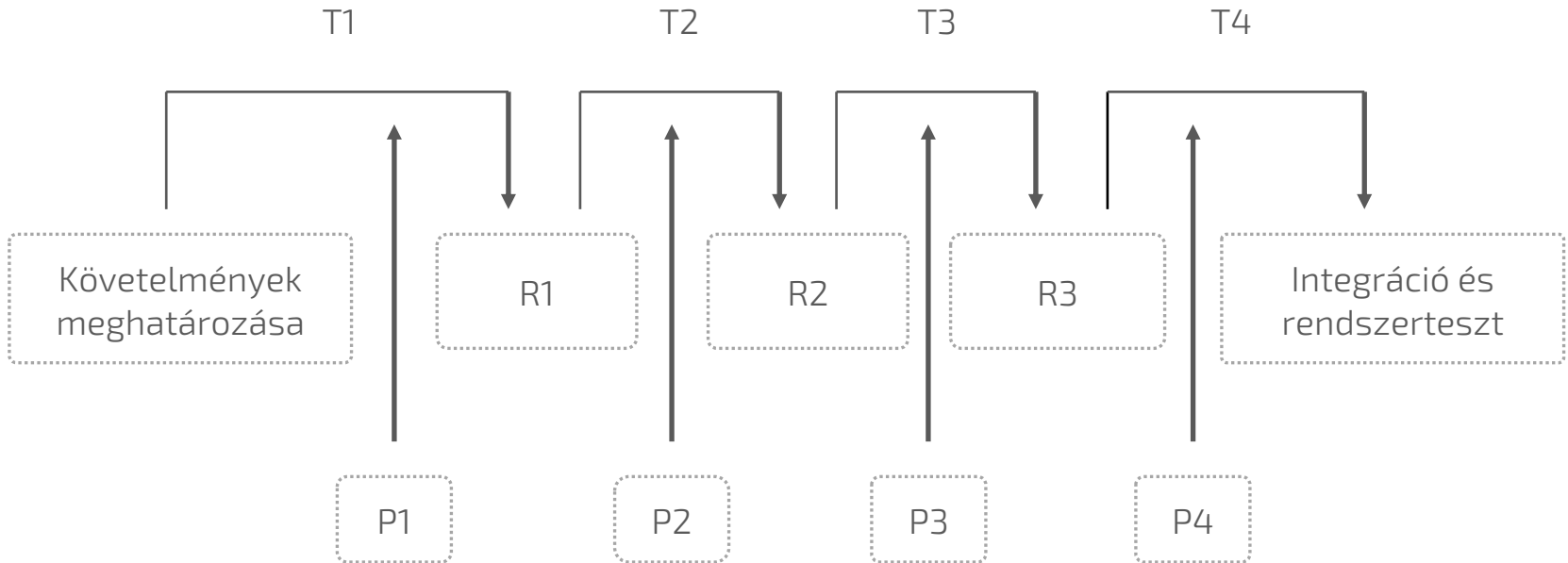
A formális rendszerfejlesztés

- A vízesés modellhez hasonló, de a fejlesztés formális matematikai eszközökkel állítja elő a futtatható programot a rendszerspecifikáció matematikai modelljéből, több transzformációs lépésen keresztül.
- Minden transzformáció során, lépésenként kell végrehajtani a tesztelést.



Formális rendszerfejlesztés

Formális transzformációk



A TRANSZFORMÁCIÓK HELYESSÉGÉNEK BIZONYÍTÁSA



Formális rendszerfejlesztés

A formális rendszerfejlesztés előnyei/hátrányai



Előnyök:

- Kritikus rendszerek esetén, ahol kulcskérdés a biztonságosság, megbízhatóság vagy védelem.
- A transzformáció és a bizonyítás részben automatizálható.



Hátrányok:

- Speciális szakértelmet igényel, meredek tanulási görbe.
- Egy rendszer kölcsönhatásait (pl. felhasználói interfész) nehéz formálisan specifikálni.
- Komplex, nagy rendszereknél ez a módszer sem eredményez jobb minőséget vagy költségmegtakarítást.



Újrafelhasználás-orientált fejlesztés



PÁZMÁNY PÉTER KATOLIKUS EGYETEM - KIEMELT FELSŐOKTATÁSI INTÉZMÉNY
INFORMÁCIÓS TECHNOLÓGIAI ÉS BIONIKAI KAR - KUTATÓ KAR

Újrafelhasználás-orientált fejlesztés

- Már létező, újrafelhasználható szoftverkomponensek egységes szerkezetbe való integrálása.
(komponens alapú rendszerfejlesztés)
- „Polcról levehető” (COTS – *Commercial Off The Shelf*) termékek felhasználása.
- Gyakran beépül a korábban ismertetett folyamatokba.
- Az elérhető komponenseket megtalálni, azokat integrálni nem egyszerű.

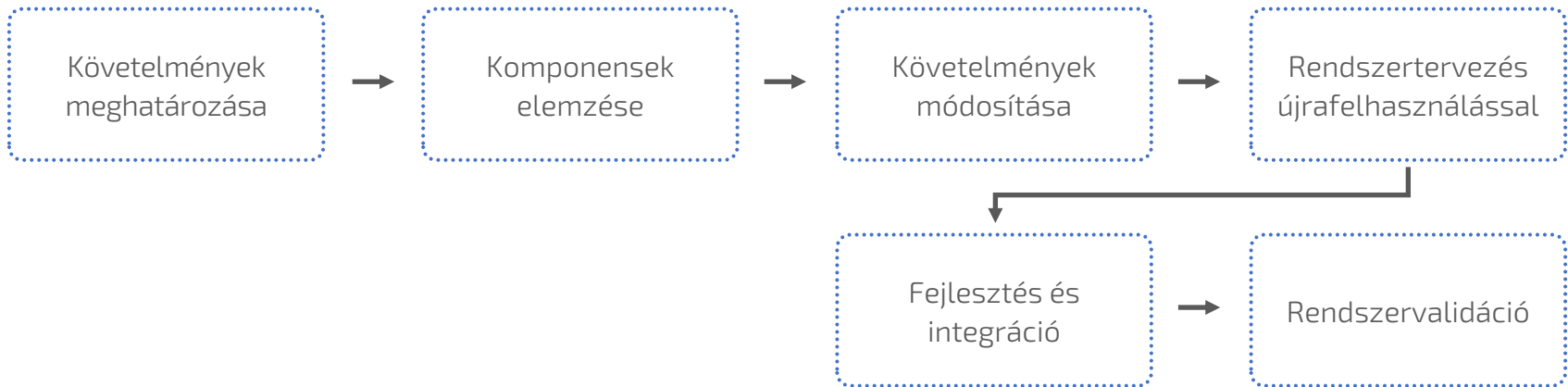
„Before software can be reusable, it first has to be usable.”
(Ralph Johnson)



Újrafelhasználás-orientált fejlesztés

A folyamat lépései:

- Követelmények meghatározása
- Komponensek keresése, elemzése
- Követelmények módosítása (*a feltárt komponensek tulajdonságainak megfelelően*)
- Rendszertervezés újrafelhasználással
- Fejlesztés és integráció
- A rendszer validációja



Folyamatiteráció – Hibrid modellek



PÁZMÁNY PÉTER KATOLIKUS EGYETEM - KIEMELT FELSŐOKTATÁSI INTÉZMÉNY
INFORMÁCIÓS TECHNOLÓGIAI ÉS BIONIKAI KAR - KUTATÓ KAR



Folyamatiteráció – Hibrid modellek

- Egy fejlesztési projekt során a rendszerkövetelmények változhatnak, ezért egyes lépéseket meg kell ismételni.
- Az iteráció az általános folyamatmodellek bármelyikével alkalmazható (*hibrid modellek*).
- Ez gyakran ellentmond a projektszervezési-, szerződéskötési elvárásoknak, amelyek jól definiált termékeket (*dokumentum vagy program*) követelnek.
- Két hibrid modellt ismertetünk:
 - **Inkrementális fejlesztés**
 - **Spirális fejlesztés**



Folyamatiteráció – Hibrid modellek Inkrementális fejlesztés



PÁZMÁNY PÉTER KATOLIKUS EGYETEM - KIEMELT FELSŐOKTATÁSI INTÉZMÉNY
INFORMÁCIÓS TECHNOLÓGIAI ÉS BIONIKAI KAR - KUTATÓ KAR

Folyamatiteráció – Hibrid modellek

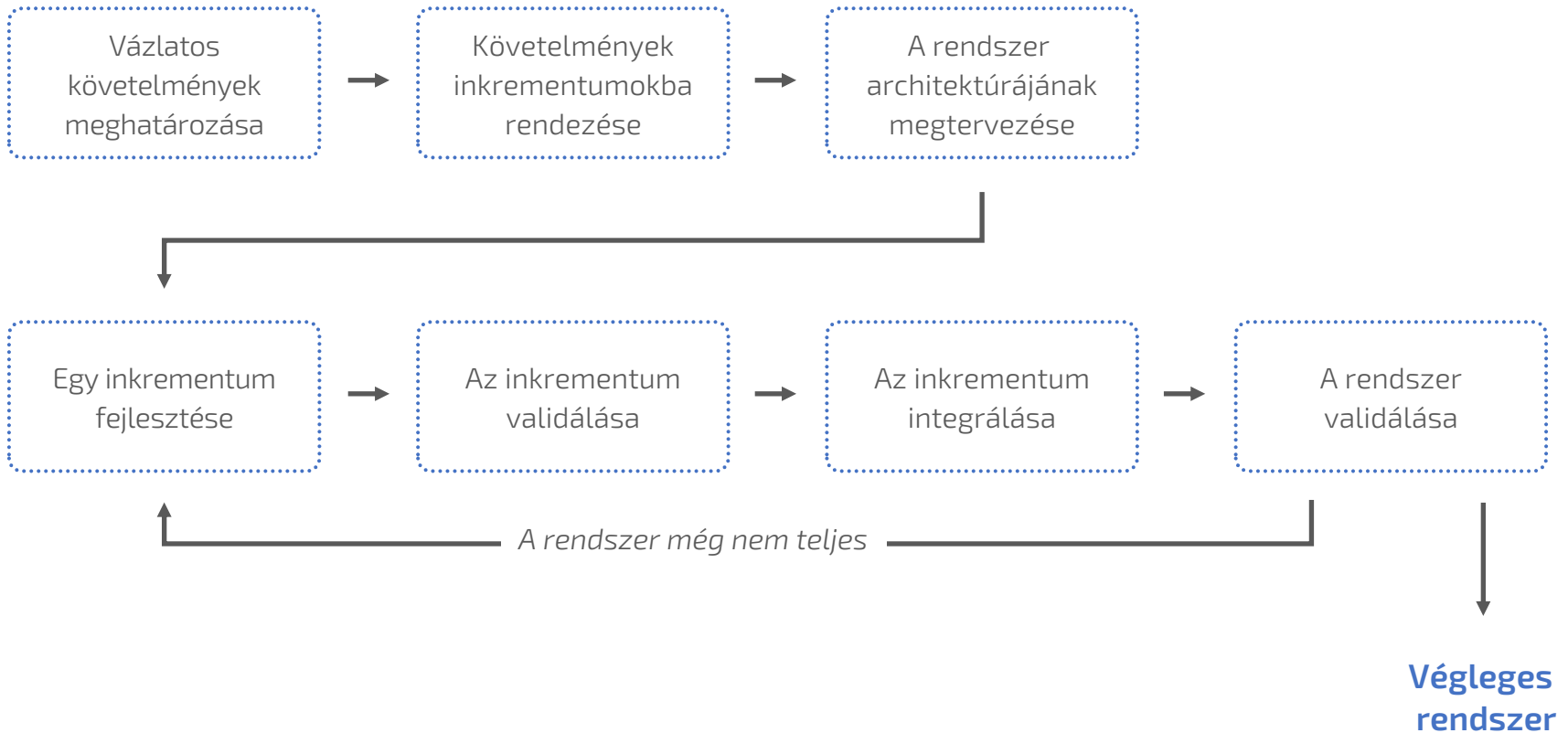
Inkrementális fejlesztés

- A szoftverfejlesztést több kisebb vízésés modellre bontja, mindegyik a funkciók egy meghatározott részét valósítja meg.
- Az első lépésekben a legfontosabb felhasználói követelményeket elégíti ki.
- Amikor egy (rész)fázisban a fejlesztés megindul, az arra vonatkozó követelményeket befagyasztja.
- Ilyen például: Agilis szoftver fejlesztés
 - **Extrém Programozás:**
kis funkcionalitással rendelkező inkrementumok készítése és átadása
 - **SCRUM:**
inkább projektszervezési módszer, de főleg szoftverfejlesztésben alkalmazzák



Folyamatiteráció – Hibrid modellek

Inkrementális fejlesztés



Folyamatiteráció – Hibrid modellek

Inkrementális fejlesztés



Előnyei:

- A felhasználó minden inkrementum átadásakor újabb, használható funkciókat kap.
- Az első inkrementumok prototípusként szolgálhatnak és segítik a későbbi inkrementumok követelményeinek meghatározását.
- Alacsonyabb a teljes projekt megghiúsulásának kockázata.
- A legfontosabb szolgáltatások hamarabb leszállíthatók, így azokat alaposabban tesztelik.



Folyamatiteráció – Hibrid modellek

Inkrementális fejlesztés



Hátrányai:

- Nagy rendszerekben sok olyan alapvető funkció van, amelyek nélkül rendszerszolgáltatás nem működhet. Ezért az első átadható inkrementum aránytalanul nagy lehet.
- Nehéz olyan inkrementumokat definiálni, amelyek ezek nélkül is használható szolgáltatást nyújtanak.
- A már leszállított inkrementumokban nagyon nehéz módosítani, ha az inkrementumok kölcsönhatása miatt szükségessé válik.



Folyamatiteráció – Hibrid modellek

Spirális fejlesztés



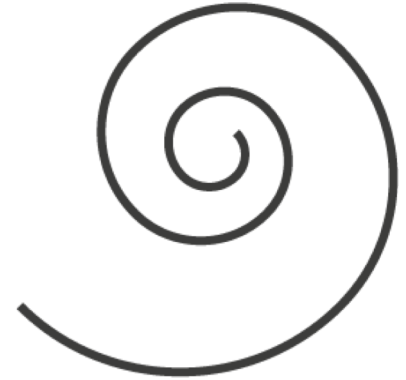
PÁZMÁNY PÉTER KATOLIKUS EGYETEM - KIEMELT FELSŐOKTATÁSI INTÉZMÉNY
INFORMÁCIÓS TECHNOLÓGIAI ÉS BIONIKAI KAR - KUTATÓ KAR



Folyamatiteráció – Hibrid modellek

Spirális fejlesztés

- A szoftverfolyamatot nem tevékenységek és esetleges visszalépések sorozataként, hanem spirálisként reprezentálja.
- A spirál minden köre a szoftverfolyamat egy fázisa.
- A spirálban nincsenek fix lépések, mint pl. specifikáció vagy tervezés, minden körben kidolgoznak egy új, az előzőnél teljesebb prototípust, amelynek „utolsó” verzióját részletes követelményspecifikációnak tekintik.
- Ennek alapján készül el a „végleges” termék.

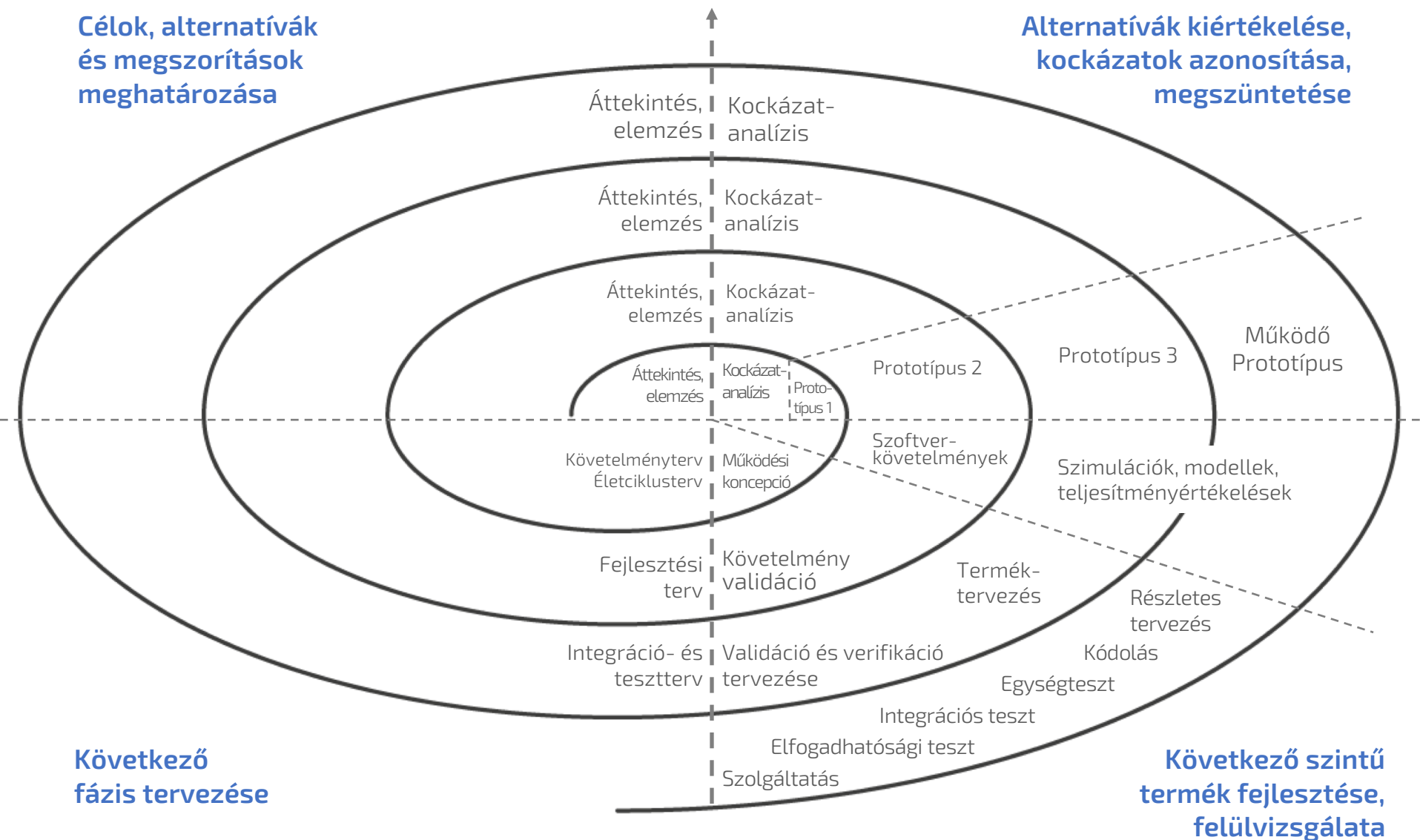


Folyamatiteráció – Hibrid modellek

Spirális fejlesztés modellje (Boehm)

Célok, alternatívák
és megszorítások
meghatározása

Alternatívák kiértékelése,
kockázatok azonosítása,
megszüntetése



Folyamatiteráció – Hibrid modellek

Spirális fejlesztés

A SPIRÁLIS MODELL SZEKTORAI

- **A célok kijelölése**
 - Az adott fázis specifikus céljainak meghatározása, elemzése
- **Kockázatok becslése és csökkentése**
 - A kockázatok elemzése, tevékenységek a kockázatok csökkentése érdekében.
- **Fejlesztés és validálás**
 - Az adott fázisnak megfelelő általános modell kiválasztása és alkalmazása.
- **Tervezés**
 - A projekt áttekintése, döntés a folytatásról, a spirál következő ciklusának tervezése



Agilis szoftverfejlesztés



PÁZMÁNY PÉTER KATOLIKUS EGYETEM - KIEMELT FELSŐOKTATÁSI INTÉZMÉNY
INFORMÁCIÓS TECHNOLÓGIAI ÉS BIONIKAI KAR - KUTATÓ KAR

Agilis szoftverfejlesztés

Változó szoftverpiac

- A szoftver a mindennapi élet része, amely sokat változott akár a 10 évvel ezelőttihez képest is.
(régi módszerek ↔ új környezet)
- A korábbi, viszonylag stabil üzleti környezet helyett ma gyorsan változnak az igények és a szoftverkészítőktől is megkövetelik, hogy gyorsan reagáljanak ezekre a változásokra.
- A szoftverfolyamatnak képesnek kell lennie arra, hogy menet közben kezelje a követelmények, a környezet és a szoftver funkcióinak változását.
- Kiszorul a piacról az a szoftverfejlesztő cég, amely nem képes a gyors, adaptív változásra.



Agilis szoftverfejlesztés

Agilis módszerek



minőség – költség – idő

- A szoftvertervezés alapproblémája, hogy megfeleljen a hármas szorításának.
- A hagyományos szoftverfolyamat ezt rengeteg dokumentációval, szigorú és merev szabályokkal igyekszik megoldani.
- A merev szabályok csak nehezen alkalmazhatók a különböző szoftvert előállító projekteken a folytonos változások közben.
- Az „agile” módszerek a dokumentáció helyett a kódra koncentrálnak, a sokféle és változó követelménynek azzal igyekeznek megfelelni, hogy **adaptív, rugalmas és kis lépésekből álló** módszereket alkalmaznak.
- Video: → [Plan-based and agile software processes – Sommerville](#) (12 perc)



Agilis szoftverfejlesztés

Szigorú módszertan – laza módszertan

SZIGORÚ MÓDSZERTAN

LAZA MÓDSZERTAN

Az információrendszerek fejlesztése:

Menedzselt, vezérelt folyamat

Véletlenszerű, alkalmazkodik a történésekhez, eseményekhez

Lineáris, szekvenciális folyamat

A folyamatok párhuzamosan, egymást átfedve zajlanak

Ismételhető, általános folyamat

Az egyes fejlesztési projektek egyediek

Racionális, célorientált és meghatározott folyamat

Megegyezéseken és kompromisszumokon alapul, gyakran „szeszélyes”



Agilis szoftverfejlesztés

Szigorú módszertan – laza módszertan

ÉRINTETT TERÜLET	AGILIS MÓDSZEREK	OPEN SOURCE MÓDSZEREK	TERVALAPÚ MÓDSZEREK
Fejlesztők	Agilis, értelmes, egy helyen dolgozó, együttműködő	Földrajzilag szétszórta, együttműködő, értelmes és agilis csoportok	Tervorientált, képzésalapú, külső tudást is felhasznál
Felhasználók	Dedikált, értelmes, egy helyen dolgozó, együttműködő, reprezentatív és elkötelezett	Dedikált, értelmes, együttműködő, elkötelezett	Elérhető az értelmes, együttműködő, elkötelezett felhasználó
Követelmények	Gyakran bekövetkező gyors változások	Gyakran bekövetkező gyors változások, folyamatosan fejlődő, soha nem befejezett	Korán megismerhető, nagyjából stabil
Architektúra	A jelen igényeire tervezett	Nyitott, a jelen igényeire tervezett	A jelen és a jövő igényeire tervezett
Méret	Kisebb csoportok, Kis rendszerek	Nagyobb, szétszórta csoportok, kisebb rendszerek	Nagy csoportok, Nagy rendszerek
Elsődleges cél, motiváció	Gyors eredmény	Kihívás	Nagy biztonság



Köszönöm a figyelmet!

Az előadásról



Az előadás a Pázmány Péter Katolikus Egyetem Információs Technológiai és Bionikai Karán meghirdetett A szoftvertechnológia alapjai című tárgy tananyagát mutatja be.

Kada Zsolt a GIRO Zrt. stratégiai és fejlesztési igazgatója.

Mérnöki képesítéseit a Torinói Műszaki Egyetemen és a Pázmány Péter Katolikus Egyetemen szerezte. Pályafutását Torinóban kutató fejlesztőként kezdte a Telecom Italia és a Politecnico di Torino közös projektjein. A pénzügyi szférában dolgozott mind banki (Erste Bank), mind beszállítói oldalon (IND). A közigazgatásban a Közigazgatási és Elektronikus Közszolgáltatások Központi Hivatalának (KEKKH) IT fejlesztési főosztályát vezette.



Kapcsolódó források



- **Vető István, A szoftvertechnológia alapjai diasor**
- **Ian Sommerville, Szoftverrendszerek fejlesztése**
 - 4. fejezet, A szoftverfolyamat
 - 25. fejezet, Az emberek menedzselése

