

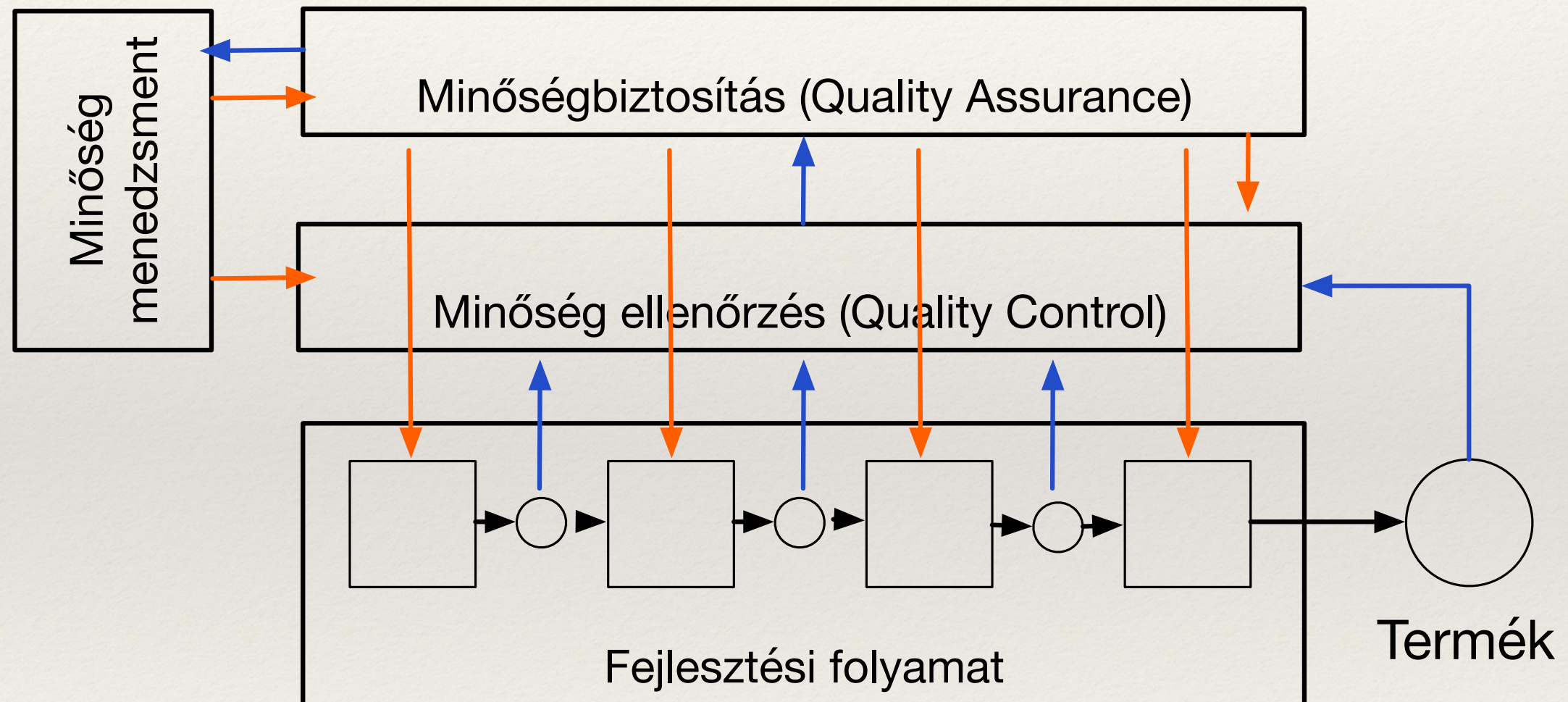
NGB_IN003_1 SZE 2017-18/2 (2)

Szoftver- minőségbiztosítás

A szoftver-
minőségbiztosítási
rendszer

A szoftver-minőségbiztosítási rendszer összetevői

Minősegbiztosítási rendszer



Minősegbiztosítási rendszer (folyt.)

	SW minősegbiztosítás	SW minőség ellenőrzés
Mi ez?	Fejlesztési folyamat minőségének fenntartása	Termékek minőségének ellenőrzése
Cél	Fejlesztési és teszt eljárások javítása	Hibák elhárítása
Megközelítés	Proaktív	Reaktív
Rendszer	Hibamegelőzési rendszer	Hibadetektáló rendszer
Felelős	Projekt team	Tesztelő team
Irány	Folyamat orientált	Termék orientált
Szkóp	Tejles fejlesztő szervezet	Termék / projekt
Eszköz	Menedzsment	Tesztelés
Életciklus	SW fejlesztési tevékenység előtt	SW fejlesztési tevékenység után
Tevékenység	Folyamat definíció, audit, training	Felülvizsgálat, tesztelés

Szoftver minőségi alapkérdések

- ❖ Hogyan hasznosítsuk a know-how-t a folyamatokban?
- ❖ Hogyan implementáljunk, specializáljunk és automatizáljunk folyamatokat különböző üzleti (szak)területeken? (Folyamatok leképezése, átvitele eszközökre)
- ❖ Hogyan garantáljuk integráltan a mérnöki ismeretek konzisztenciáját, a szolgáltatások minőségét?

Szoftver min.bizt. rendszer – Szoftver min.bizt. architektúra

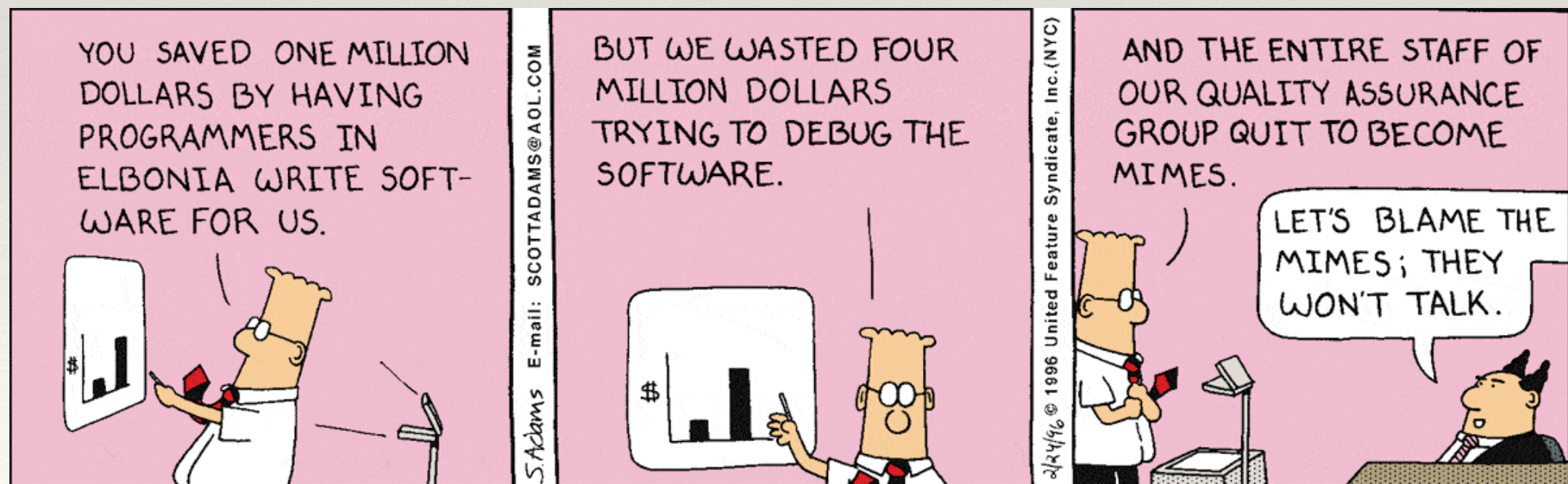
- ❖ A szoftver-minőségbiztosítási rendszer különböző összetevőket kombinál a **szoftver hibák hatásának elfogadható szintre csökkentésére**.
- ❖ A szoftver min.bizt. rendszer összetevői (6 kategória):
 - ❖ elő-projekt komponensek
 - ❖ élet ciklus ellenőrző tevékenységek
 - ❖ hiba-megelőzési és -javítási tevékenységek
 - ❖ szoftver minőség-menedzsment
 - ❖ szabványosítás, tanúsítás
 - ❖ humán összetevők

Elő-projekt tevékenységek

- ❖ Szerződés átvizsgálás
 - ❖ követelmények tisztázása
 - ❖ proj. ütemezés és erőforrás becslések átvizsgálása
 - ❖ munkaerő értékelése
 - ❖ projekt kockázatok értékelése
- ❖ Minőségbiztosítási tervezés
 - ❖ minőségi célok, kritériumok meghatározása
 - ❖ átvizsgálások, tesztek, verifikációs és validációs tevékenységek listája

Min. bizt. élet ciklus tevékenységek

- ❖ Felülvizsgálatok (review)
- ❖ Szakértői vélemények
- ❖ Szoftver tesztelés
- ❖ Karbantartás
- ❖ Harmadik féltől származó megoldások értékelése



Felülvizsgálatok, átvizsgálások

- ❖ Formális felülvizsgálatok
 - ❖ -> formális jóváhagyás (következő fázisba lépésre)
 - ❖ rögzített forma, résztvevők (megrendelő részéről is)
 - ❖ vezető szakemberek (seniors)
 - ❖ értékelés és meghatározott akciók
- ❖ Átvizsgálások (peer review, inspection, walkthrough)
 - ❖ kisebb egységekre vonatkozóan
 - ❖ fejlesztők (kollegiális viszony, reciprocitás)

Szoftvertesztelés

- ❖ Szoftver futtatása
- ❖ Tesztesetek (scenáriók) végrehajtása
- ❖ Különböző kiterjedés
 - ❖ modulok (szoftver egységek)
 - ❖ integrált komponensek
 - ❖ teljes rendszer
- ❖ Regressziós tesztek (tesztek újra végrehajtása)
- ❖ Célok
 - ❖ hibák megtalálása
 - ❖ modulok, integrálás jóváhagyása

Szoftvertesztelés (folyt.)

- ❖ Tesztek végrehajtása
 - ❖ kézi
 - ❖ automatizált
- ❖ Dokumentumok
 - ❖ teszt tervek (tesztesetek, konfiguráció)
 - ❖ teszt riport (eredmények, lefedési mértékek)
- ❖ Független tesztelés előnyei

Szoftver-karbantartás

- ❖ Javító karbantartás - kód és dokumentáció hibák javítása
- ❖ Adaptív karbantartás - új feltételekhez igazítás a szoftver termék alapvető megváltoztatása nélkül
- ❖ Kibővítő karbantartás - új funkcionalitás hozzáadása a meglévő szoftverhez
- ❖ A karbantartásra az eredeti fejlesztéshez hasonlóan érvényes a szoftver-minőségbiztosítási rendszer

Külső források minősegbiztosítása

- ❖ Beszállítók, alvállalkozók, COTS (dobozos) szoftverek
- ❖ Gazdaságossági és műszaki megfeleléségi kérdések
- ❖ Harmadik fél saját minősegbiztosítási rendszere
- ❖ Szabványok
- ❖ Konformitási tesztek

Hiba-megelőzési és tökéletesítési tevékenység

- ❖ Szoftverhibák (fault) megelőzése >
 - ❖ Meghibásodási ráta csökkentése (termelékenység fenntartása mellett)
- ❖ Eljárás- és munkautasítások
- ❖ Sablonok, ellenőrzőlisták
- ❖ Képzés, minősítés
- ❖ Konfiguráció menedzsment
- ❖ Preventív lépések
- ❖ Dokumentum-kontrol

Szoftver-min. bizt. menedzsment

- ❖ Projekt előrehaladásának ellenőrzése
 - ❖ erőforrás felhasználás
 - ❖ határidők
 - ❖ kockázatkezelés
 - ❖ költségvetés
- ❖ Szoftver minőségi mértékek meghatározása
 - ❖ fejlesztési tevékenység minősége
 - ❖ fejlesztés produktivitása
 - ❖ szoftver hiba sűrűség
- ❖ Minőség biztosítási költségek kezelése

Menedzsment szerepe a szoftver min.bizt.-ban

- ❖ Minőség-politika meghatározása
- ❖ A minőség-politika implementálásának követése
- ❖ Erőforrások biztosítása az implementációhoz
- ❖ Megfelelőségi és értékelési folyamatok létrehozása

Minősegbiztosítási szervezet létrehozása

- ❖ Szempontok:
- ❖ Szervezeti megfontolások
 - ❖ Ügyfélkör (fejlesztés, karbantartás)
 - ❖ Termékkör kiterjedése
 - ❖ Szervezet mérete
- ❖ Projekt megfontolások
 - ❖ A szoftver komplexitása
 - ❖ Fejlesztési technológiai tapasztalat
- ❖ Személyzeti megfontolások
 - ❖ Szakmai szint
 - ❖ Csapattagok ismertsége

Szoftver-minőségbiztosítás az élelciklusban

Életciklus modellek

- ❖ Szoftver életciklus modell = szoftver (fejlesztési) folyamat modell
- ❖ A szoftver fejlesztésével kapcsolatos koncepciók, módszerek konzisztens elrendezése
- ❖ Fő tevékenységek és végrehajtási sorrendjük (fázisok, mérföldkövek)
- ❖ Szoftver minősegbiztosítási tevékenységek a mérföldkövekhez kapcsolódóan

Életciklus modellek (folyt.)

- ❖ A fő szoftverfejlesztési tevékenységek:
 - ❖ Követelmény elemzés és definiálás
 - ❖ Rendszer (architektúra) tervezés
 - ❖ Részletes (program) tervezés
 - ❖ Implementálás (kódolás)
 - ❖ Tesztelés
 - ❖ Üzembehelyezés
 - ❖ Karbantartás

Folyamat-modellek haszna

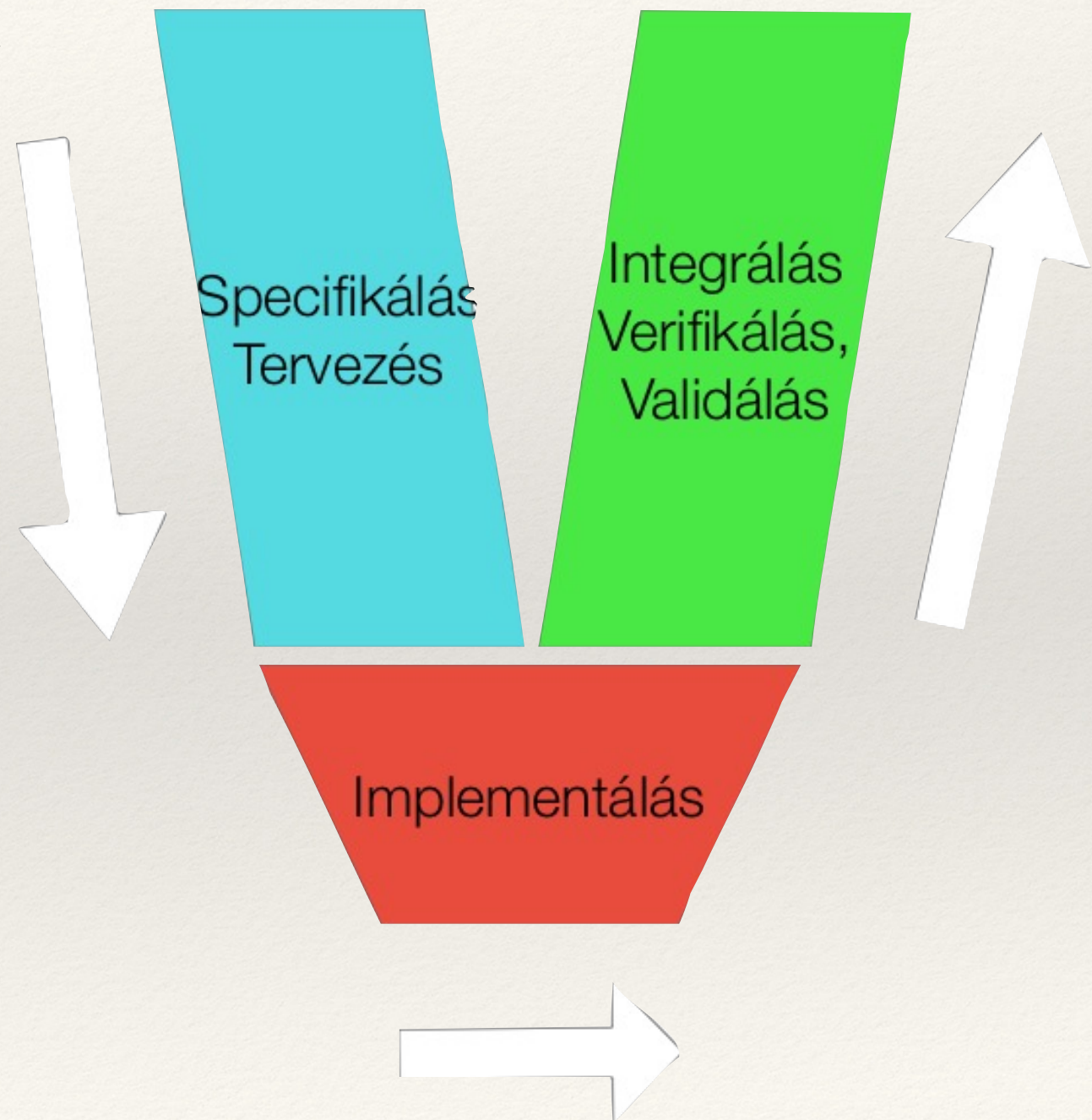
- ❖ Emberi megértés és kommunikáció elősegítése
- ❖ Folyamatok javításának támogatása
- ❖ Folyamat menedzsment támogatása
- ❖ Támogatja a folyamat végrehajtásának és / vagy irányításának automatizálását

Néhány életciklus modell

- ❖ Vízesés modell
- ❖ V-modell
- ❖ Spirál modell
- ❖ Inkrementális és iteratív modellek
- ❖ Agilis modellek

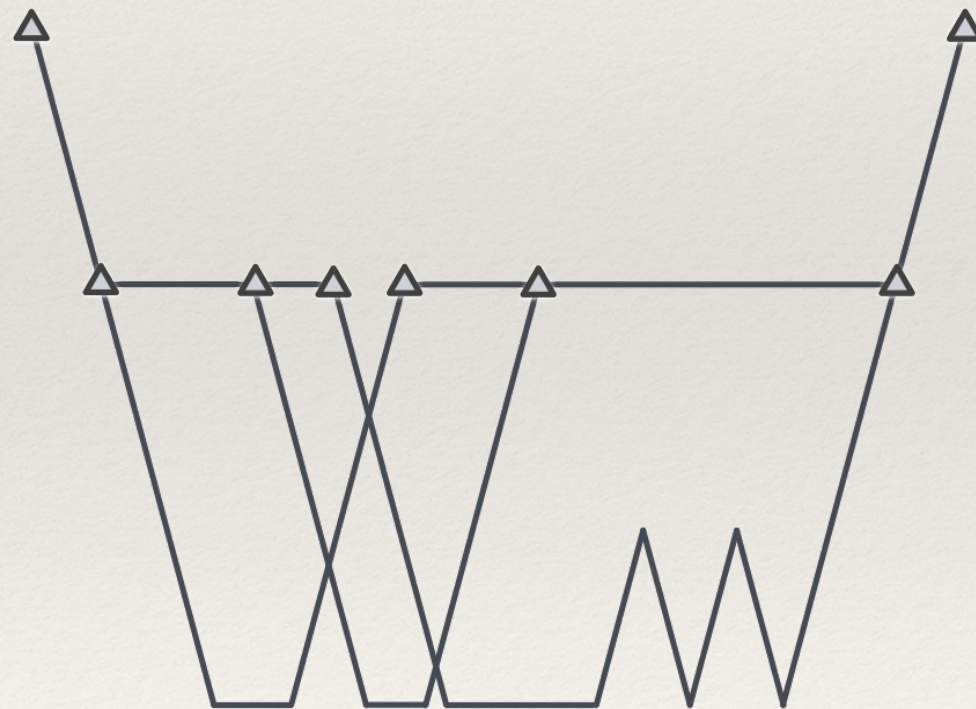
A V-modell (példa)

- ❖ A vízesés modell egy változata, iterálási lehetőséggel
- ❖ Két ág:
 - ❖ specifikálás és tervezés (a rendszer lebontása)
 - ❖ integrálás és validálás
- ❖ Széleskörű alkalmazás biztonság kritikus rendszerek fejlesztésekor



A V-modell (folyt.)

- ❖ A teljes életciklus több elemi V ciklusból épülhet fel
 - ❖ alrendszerek, komponensek, verziók
 - ❖ párhuzamos és szekvenciális fejlesztés



A minősegbiztosítási tevékenység intenzitása

- ❖ Projekt faktorok
 - ❖ Méret
 - ❖ Technikai komplexitás és nehézség
 - ❖ Szoftver újra felhasználás
 - ❖ Meghibásodások kimenetelének súlyossága
- ❖ Team faktorok
 - ❖ Csapattagok szakmai kvalitása
 - ❖ A csapat ismeretei a projektről és a szakterületről
 - ❖ Szakmai támogatás rendelkezésre állása
 - ❖ Csapat összeszokottsága

Verifikálás és validálás

- ❖ IEEE 610.12-1990
- ❖ **Verifikálás:** Egy rendszer vagy komponens értékelése abból a szempontból, hogy kielégíti-e a **fázis kezdetekor** meghatározott feltételeket, követelményeket.
- ❖ **Validálás:** Egy rendszer vagy komponens értékelése abból a szempontból a **fejlesztés lezárultával**, hogy kielégíti-e a specifikált követelményeket.

Verifikálás és validálás (folyt.)

- ❖ Azaz:
- ❖ **Verifikálás: konzisztencia az előző fázisban létrehozott termékkel (design), feltételezve, hogy az megfelelő volt.**
A vizsgálat nem veszi figyelembe az eredeti (megrendelői) követelményeket!
- ❖ **Validálás: a megrendelő érdekeinek figyelembe vétele, megfelelés az eredeti követelményeknek.**
(Megrendelői elégedettség maximalizálása)

Verifikálás és validálás módszerei

- ❖ Felülvizsgálat, átvizsgálás (review, inspection)
- ❖ Elemzés (modellezés, szimuláció, számítások)
- ❖ Demonstráció (működő rendszer, spec. műszerezés nélkül)
- ❖ **Tesztelés**
- ❖ Tanúsítás (jogi, ipari szabványoknak megfelelés, külső tanúsító / hatóság)

A szoftver min.bizt. hiba-eltávolító hatékonysága és költsége

- ❖ Hibák forrásának (keletkezési helyének) megoszlása
- ❖ Felmérésekből származó statisztikák szerint:

Szoftver fejlesztési tevékenység	Hiba megjelenés aránya
Követelmény specifikálás	15 %
Tervezés	35 %
Kódolás, integrálás	40 %
Dokumentálás	10 %

A szoftver min.bizt. hiba-eltávolító hatékonysága

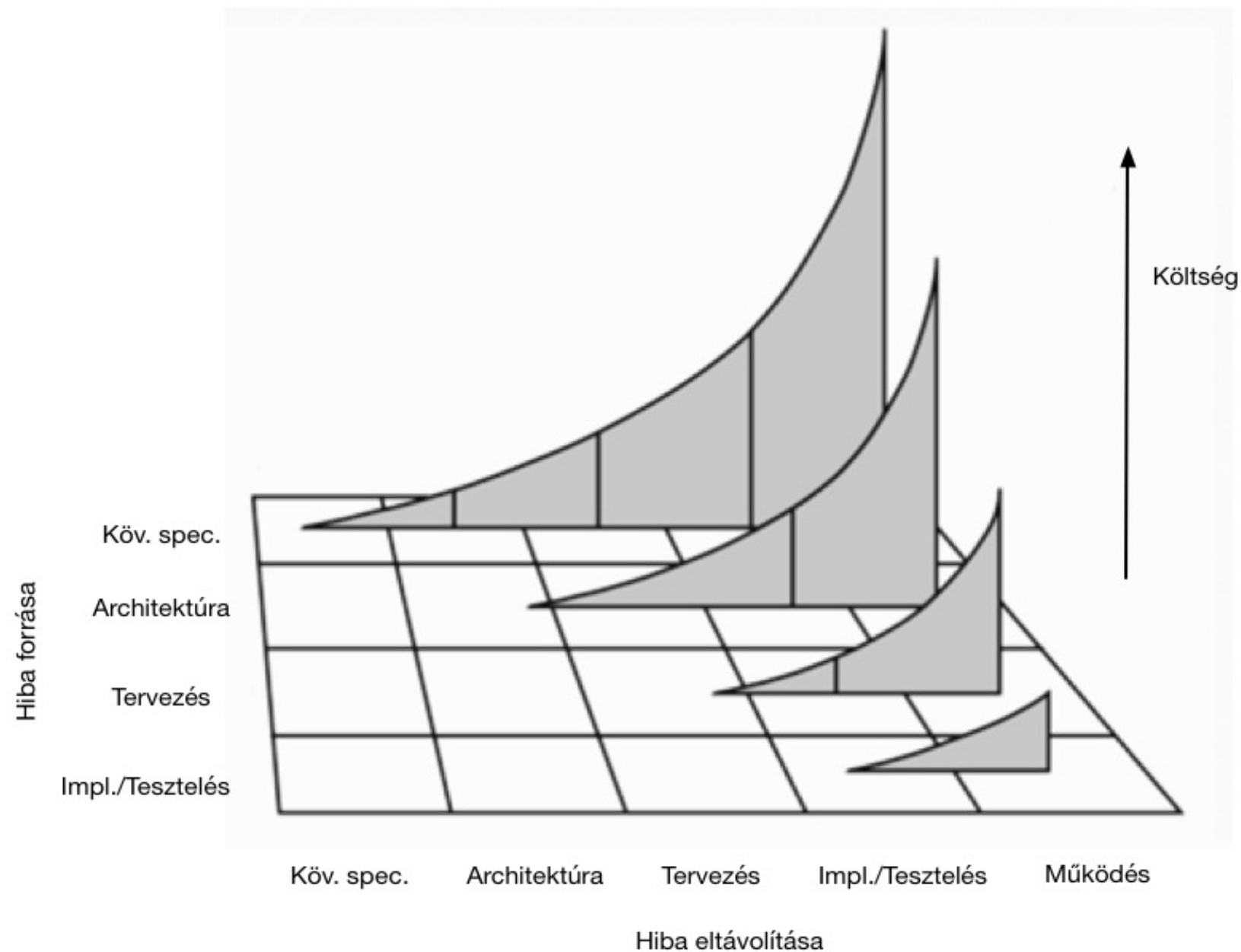
- ❖ Minden szoftver min.bizt. tevékenység kiszűri a hibák egy bizonyos részét
- ❖ Az eltávolított hibák száma (kicsit) kevesebb, mint a detektált hibák száma (nem megfelelő / nem hatékony hiba javítás)
- ❖ Minden következő min.bizt. tevékenység a hibák kombinációjával szembesül (maradék + új)
- ❖ Az akkumulált hibák eltávolításának mértéke feltételezhetően 40% körüli

A hiba-eltávolítás költsége

- ❖ A hiba-eltávolítás költsége tevékenységenként (fázisonként) eltérő
- ❖ A költség jelentősen emelkedik a folyamat továbbhaladtával
- ❖ Szekvenciális fázisokat feltételezve:

Szoftver fejlesztési fázis	Hiba eltávolítás átlagos relatív költsége
Követelmény specifikálás	1,0
Tervezés	2,5
Egységtesztelés	6,5
Integrációs tesztelés	16,0
Rendszer teszt	40,0
Működés közben, ügyfél által	110,0

A hiba-eltávolítás költsége (folyt.)



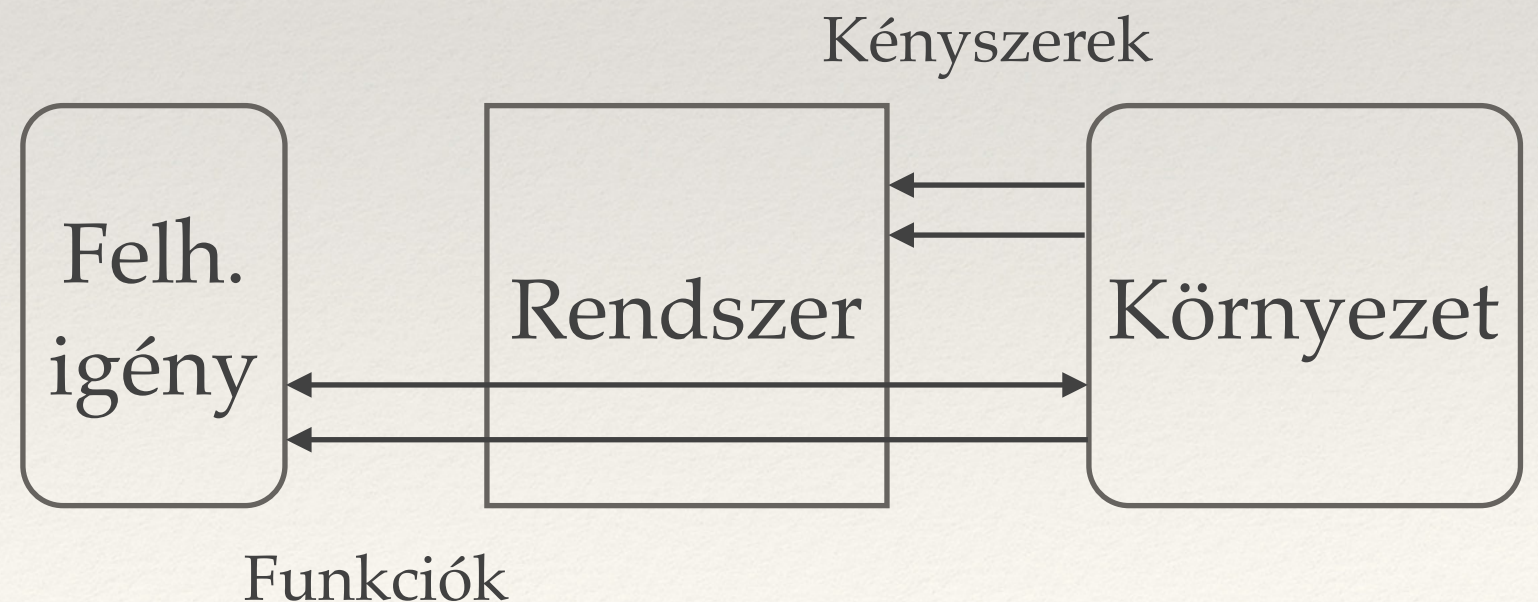
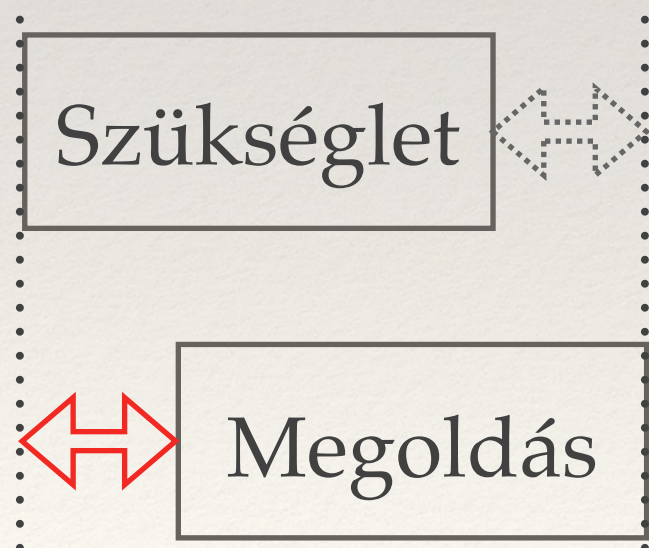
Követelmények és minőség

Követelmény típusok

- ❖ Funkcionális követelmények
 - ❖ A rendszer funkciójából fakadó viselkedésének eredményét meghatározó követelmény.
- ❖ Minőségi követelmény
 - ❖ Funkcionális követelménnyel nem lefedett, a minőségre vonatkozó követelmény.
- ❖ Kényszer, korlátozás (constraint)
 - ❖ A megoldási (tervezési) teret korlátozó, a funkcionális és minőségi követelmények eléréséhez szükséges követelmény.

Követelmények kialakítása

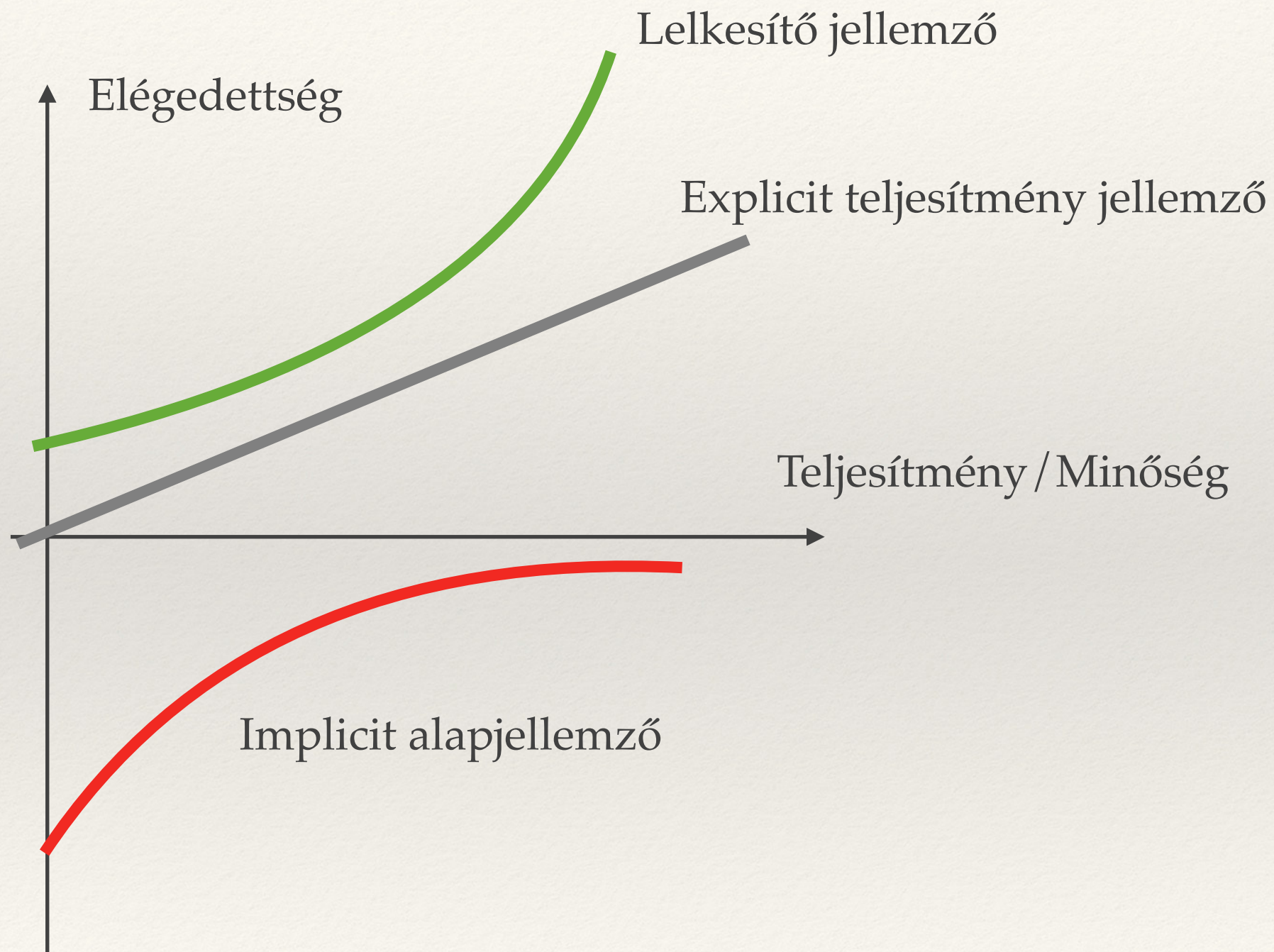
- ❖ A követelmények forrása
 - ❖ Érintettek (stakeholders)
 - ❖ Dokumentumok
 - ❖ Meglévő rendszerek (legacy systems)
- ❖ Szükségletek és megoldások



Ügyfél értékek

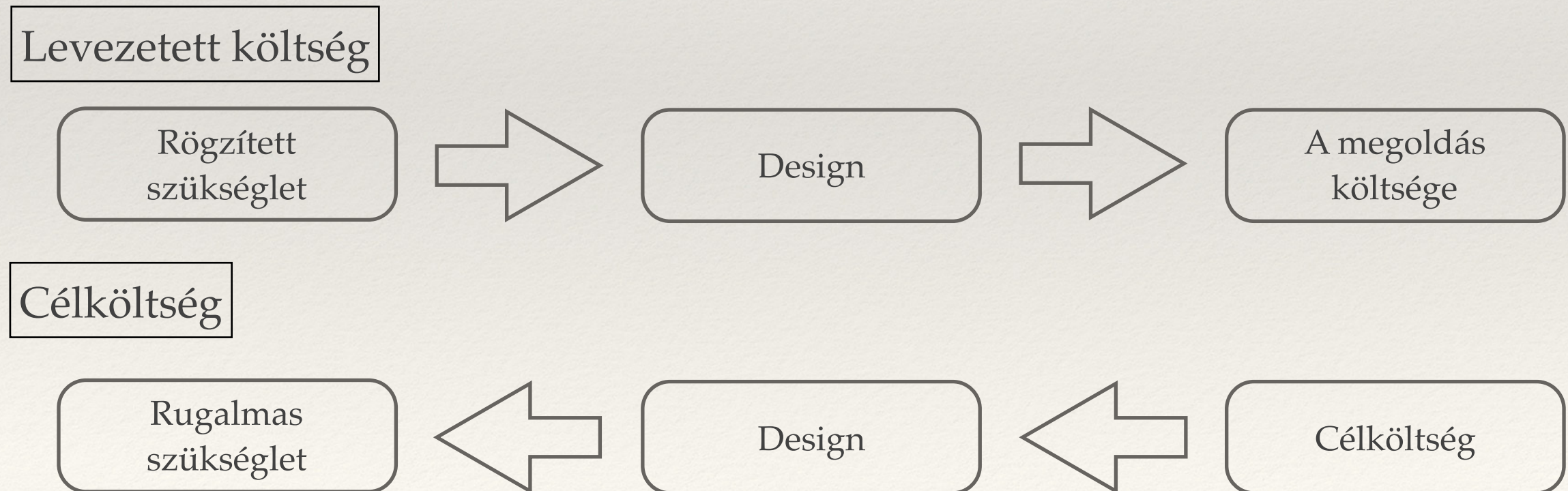
- ❖ Adott követelmény fontossága, értéke az ügyfél számára
- ❖ Kano modell
 - ❖ Magától értetődő (alap) követelmények (nem teljesülésük elégedetlenséget okoz) (dissatisfier)
 - ❖ Expliciten megadott követelmények (teljesülésük elégedettséget okoz) (satisfier)
 - ❖ Nem várt kedvező tulajdonságok (meglétük lelkesítőleg hat) (delighter)

Ügyfél értékek (folyt.)



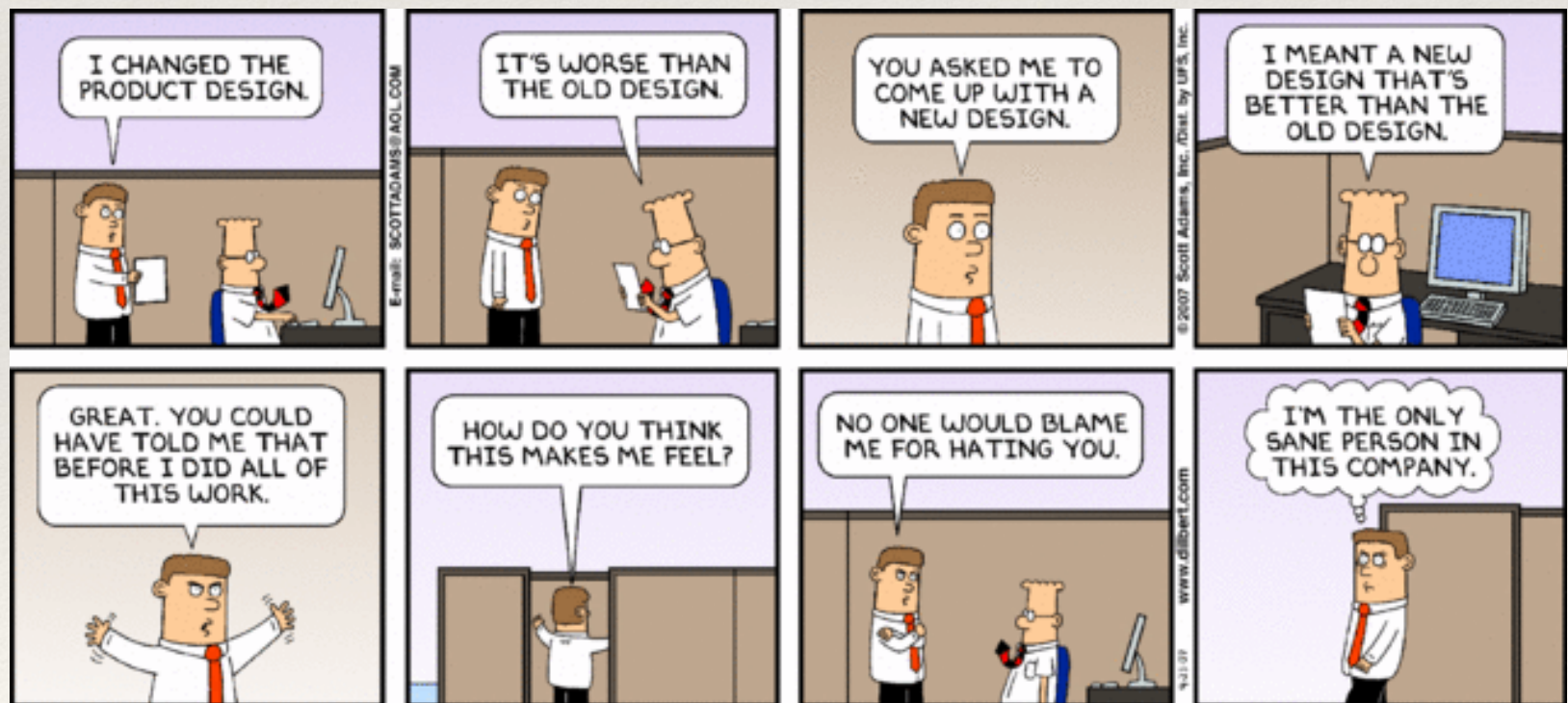
Költség mint bemeneti követelmény

- ❖ A fejlesztési költség saját maga is egy teljesítmény jellemző (cél) a technikai jellemzők mellett
- ❖ A célköltséghez való konvergencia biztosítandó a fejlesztési folyamat során



Követelmények minőségi jellemzői

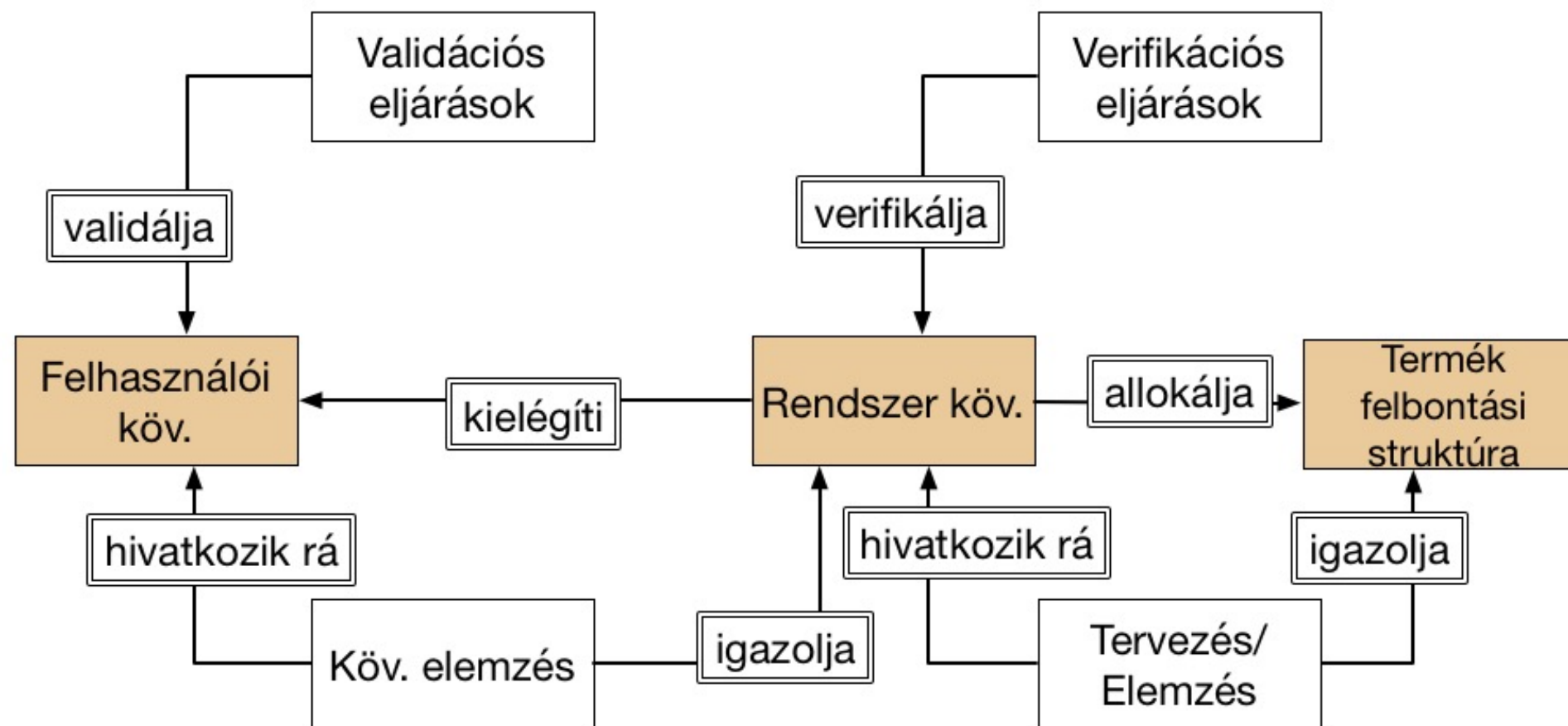
- ❖ Jóváhagyott
- ❖ Egyértelmű
- ❖ Érvényes, időszerű (up-to-date)
- ❖ Korrekt
- ❖ Konzisztens
- ❖ Verifikálható
- ❖ Megvalósítható
- ❖ Nyomonkövethető
- ❖ Teljes
- ❖ Érthető



Követelmény menedzsment

- ❖ Követelmény-történet (változások) monitorozása
- ❖ Követelmények (egymást követő halmazok) verifikálása és validálása
 - ❖ követelmények viszonya egymáshoz
 - ❖ követelmények leképzése a tervezés során
 - ❖ követelmények validálhatósága a terméken
- ❖ A szoftver verifikálásának és validálásának követése követelményekhez viszonyítva

Követelmény menedzsment rendszer



Követelmény menedzsment rendszer (pl.)

❖ Rational DOORS

The screenshot displays the Rational DOORS software interface. The title bar reads "DOORS: Formal module 'High Level Requirements' current 3.1". The menu bar includes "Module", "Edit", "View", "Object", "Attribute", "Column", "Link", "Table", "Tools", "User", "Kitchen", "Word", and "Help". The toolbar contains various icons for file operations, editing, and viewing. Below the toolbar, there are dropdown menus for "Rich traceability" and "All levels".

Identifier	High Level Requirements	Rich Traceability	Contributing Requirements
HL126-Com	3 Functional requirements		
HL145-Com	3.1 General		
HL198-Req	Main safety objectives of an <u>interlocking</u> are to:		
HL9-Req	- prevent collision between <u>trains</u>		
HL10-Req	- prevent derailment of <u>trains</u> (e.g. at incorrectly set <u>points</u> or inadequately <u>locked facing points</u>)		
HL12-Req	- prevent collision with non-railway traffic (e.g. cars, trackside workers, pedestrians etc.) and railway infrastructure (e.g. buffer stop etc.).		
HL11-Req	- give a safe <u>movement authority</u>		SI4305-Req SI4259-Req SI2687-Req
HL14-Req	The <u>interlocking</u> shall maintain the integrity of the <u>route</u> once the <u>movement authority</u> has been given.		SI4308-Req SI4307-Req
HL15-Req	No <u>movement authority</u> issue shall be permitted unless the <u>route</u> is <u>set</u> and <u>locked</u> .		SI589-Req
HL16-Req	Once the <u>route</u> has been <u>set</u> and <u>locked</u> , and the <u>movement authority</u> has been issued, the <u>interlocking</u> shall not permit any <u>points</u> in the <u>route</u> to be moved or conflicting <u>movement authority</u> to be issued until the <u>train</u> has used the <u>route</u> , or until it has been established that the <u>train</u> will not use the <u>route</u> .		
HL179-Req	The <u>interlocking</u> shall allow a <u>movement authority</u> only for so long as the conditions, which allowed it to be issued are maintained.		SI2409-Req
HL1173-Req	The loss of the necessary conditions after a <u>movement authority</u> has been issued shall result in the immediate withdrawal of <u>movement authority</u> .		SI592-Req SI4257-Req SI2655-Req
HL1174-Com	3.2 Basic <u>interlocking elements</u>		
HL91-Co	3.2.1 <u>Points</u>		

User name: CT5 Default Link Module: DOORS Links

Taskbar: Start | DOORS: Project Manager ... | Presentation1 | DOORS: Formal modu... | 4:36 PM