

NGB_IN003_1 SZE 2017-18/2 (5)

Szoftver- minőségbiztosítás

Szoftver-minőségbiztosítási
szabványok

Szabványok

- ❖ Termék jellemzők vagy folyamatok rögzített (elismert szervezet által jóváhagyott) normáit, követelményeit tartalmazó dokumentum
- ❖ Szolgáltatásokra vonatkozó műszaki követelmények, vizsgálatok, minősítési feltételek és meghatározások
- ❖ A szabványosítás a felhasználó és a fogyasztó érdekében végzett szabályozó, egységesítő tevékenység

Szabványok alkalmazása

- ❖ Szabvány alkalmazása:
 - ❖ jogilag kötelező alkalmazás (de jure)
 - ❖ ajánlott, önkéntes
 - ❖ piaci hatásra dominánsan alkalmazott (de facto)
- ❖ Nyílt szabványok
 - ❖ nyilvánosan hozzáférhető, és a szabványt bárki használhatja, a szabványhoz fűződő szellemi tulajdonjogok egy non-profit szervezet birtokában vannak

Szoftver-minőség és szabványok

- ❖ Teljes szoftverek viszonylag ritkán felelnek meg szabványoknak
- ❖ program működés, viselkedés nehezen szabványosítható
- ❖ fejlesztési folyamat szabványosítható, de nem megfelelő szabványosítás csökkent(het)i a termelékenységét

A szabványosítás lehetséges előnyei

- ❖ Fejlesztési és karbantartási tevékenységek végzése a legmagasabb professzionális szinten
- ❖ Jobb koordináció a fejlesztő csapatok között
- ❖ Jobb kooperáció a külső szereplőkkel
- ❖ Megértés javítása a megrendelő és a fejlesztő szervezet között
- ❖ => Szakmai és menedzsment munka minőségének javulása
- ❖ => Tanúsítás és audit alapjainak megteremtése

A szabványosítás lehetséges előnyei (folyt.)

- ❖ Nyereségek szabványok alkalmazásával:
 - ❖ Specialitások, különlegességek elkerülése (pl. progr. nyelvek)
 - ❖ Megismételhetőség (pl. komplex ellenőrzési eljárások)
 - ❖ Konszenzus, tudás (pl. szoftver metrikák)
 - ❖ Megrendelő védelme (min.bizt. szabványok)
 - ❖ Professzionális fegyelem (pl. V&V eljárások)

Jogi vonatkozások

- ❖ De jure szabvány alkalmazás
 - ❖ törvény(ek) írják elő a megfelelést
 - ❖ Szoftverek viszonylag szűk körére - biztonság kritikus alkalmazások
- ❖ Önkéntes szabvány alkalmazás a "megfelelő gondosság" alátámasztására
- ❖ Termékhibák visszavezetése a szabvány hibájára



Szabvány típusok

- ❖ Szoftver-minősegbiztosítási szabványok (tanúsítási, értékelési módszerek)
 - ❖ szervezet min. bizt. rendszerének kialakítása
 - ❖ beszállítók minősítése
- ❖ Fejlesztési folyamat szabványok
 - ❖ szoftver folyamat megvalósítása



Szabvány típusok (folyt.)

	Minőség menedzsment szabványok	Projekt folyamat szabványok
Alkalmazó	Menedzsment, min. bizt részleg	Fejlesztő teamek
Fókusz	Minőségbizt. rendszer, infrastruktúra, köv. meghatározás	Fejlesztési módszerek
Szabványok tárgya	Mit kell elérni?	Hogyan kell csinálni?
Szabványok célja	Fejlesztési folyamathoz kapcsolódó min. bizt. tevékenység ellenőrzése	Szoftver termékek minőségének biztosítása

Szabványügyi szervezetek

- ❖ ANSI: American National Standards Institute
- ❖ IEC: International Electrotechnical Commission
- ❖ IEEE: Institute of Electrical and Electronics Engineers Computer Society Software Engineering Standards Committee
- ❖ ISO: International Organization for Standardization
- ❖ CENELEC: Comité Européen de Normalisation Électrotechnique
- ❖ ITU-T (CCITT): International Telegraph Union
- ❖ Nemzetközi szabványokat gyakran adoptálják nemzeti szabványként.

A szoftver-min.bizt-ban alkalmazott szabványok

- ❖ Számítás tudományi szabványok (pl. IEEE Std 610.12:1990)
- ❖ Minőségbiztosítási / minőség menedzsment szabványok (pl. ISO 9000-3)
- ❖ Projekt menedzsment szabványok (pl. IEEE Std 1058.1-1987)
- ❖ Rendszer mérnökségi szabványok (pl. ISO / IEC WD 15288)
- ❖ Biztonsági szabványok (pl. IEC 61508)
- ❖ Termék szabványok (pl. ISO / IEC 14598)
- ❖ Folyamat szabványok (pl. ISO / IEC 12207:1995)
- ❖ Jellemzően a **folyamatok**, nem a termékek konzisztens minősége biztosítható.

Szoftver-minőségbiztosítási szabványok

- ❖ **Tanúsítási szabványok:**
 - ❖ Fejlesztő szervezet számára lehetővé teszi, hogy a szoftvertermék vagy karbantartási szolgáltatás **minőségének konzisztens fenntartásának képességét** demonstrálja.
 - ❖ Megállapodott értékelési keret a minőségbiztosítási rendszerre vonatkozóan.
 - ❖ Lehetővé teszi a minőségbiztosítási rendszer javítását.

Szoftver-minőségbiztosítási szabványok

- ❖ **Értékelési szabványok:**
 - ❖ **Önértékelési eszköz a fejlesztési projekt minőségére vonatkozóan.**
 - ❖ **Fejlesztési és karbantartási folyamatok javítása**
 - ❖ **Potenciális beszállítók értékelése**

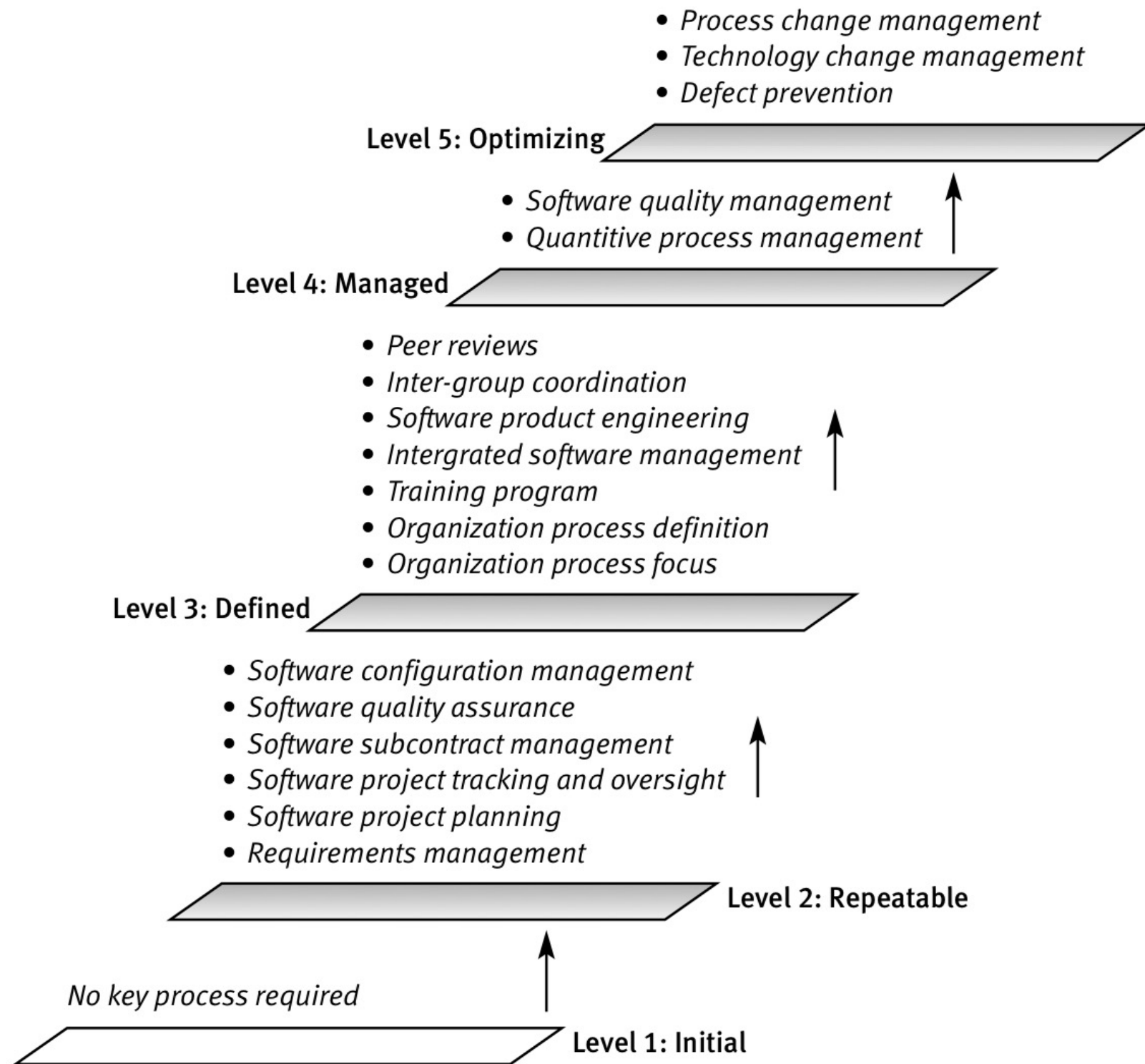
Példa: ISO 9000-3

- ❖ A szabvány alapelvei:
 - ❖ Fókuszálás a megrendelőre (követelmények meghat.)
 - ❖ Folyamat megközelítés (tevékenységek és erőforrások folyamat szerű menedzselése)
 - ❖ A menedzsment rendszer-szerű megközelítése
 - ❖ Folytonos javítás fontossága
 - ❖ Tényszerű megközelítés a döntések meghozásakor

Képesség érettségi modell - CMM

- ❖ Capability Maturity Model (Carnegie Mellon University SEI)
- ❖ Szervezet képessége a minőség és termelékenység kontrolálására
- ❖ Öt képesség érettségi szint
- ❖ Generikus a folyamat-területekre (mit és nem hogyan)
 - ❖ tetszőleges életciklus modell
 - ❖ tetszőleges fejlesztési módszer és eszközök

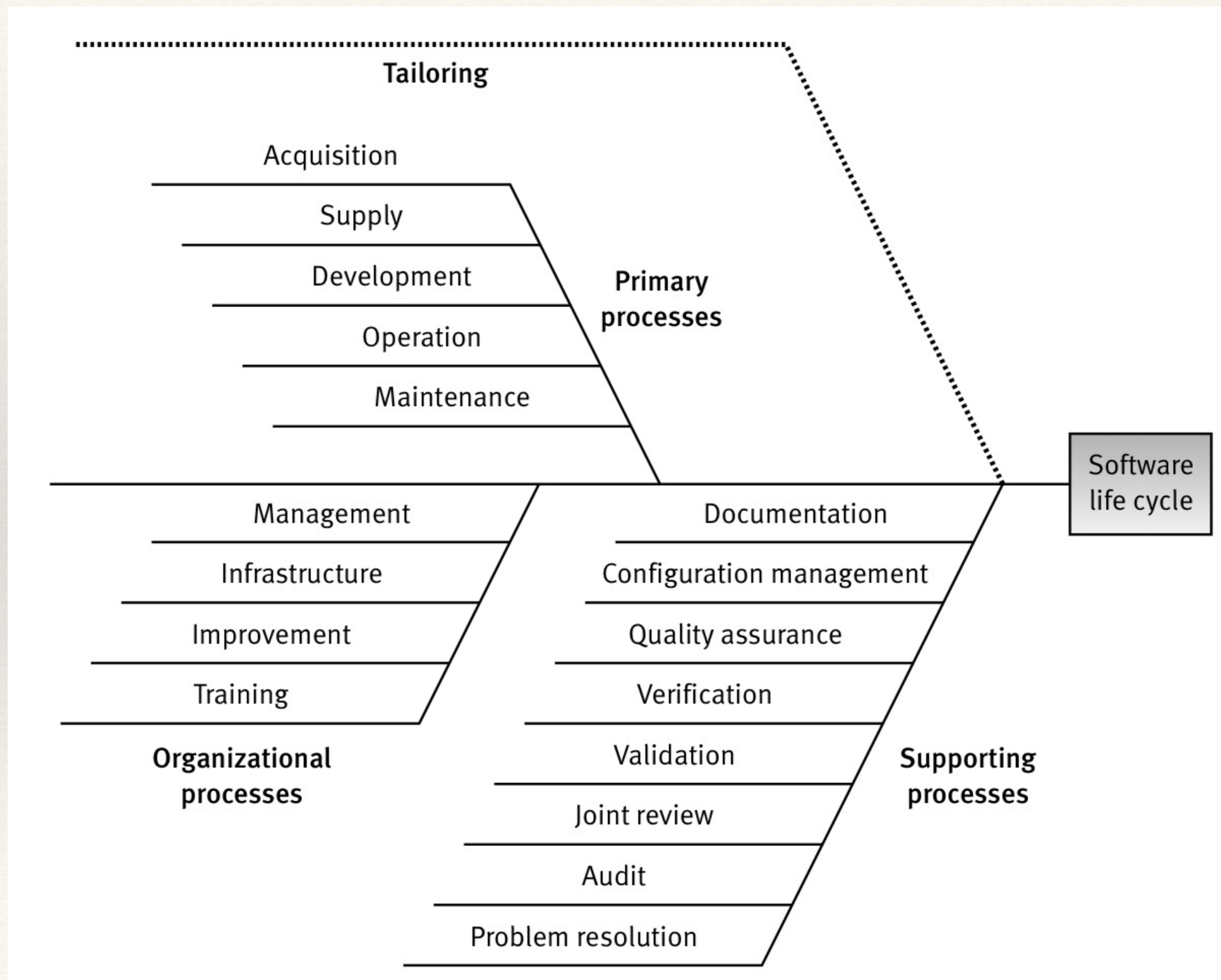
CMM szintek



Szoftver fejlesztési projekt szabványok

- ❖ Módszerek, módszertanok előírása a fejlesztésre és karbantartásra
 - ❖ projekt folyamat eljárások
 - ❖ teamek és külső szereplők koordinálása
 - ❖ szoftver-technológiai (SE) szabványokként működnek
 - ❖ pl. IEEE Software Engineering Standard Collection

Példa: IEEE 12207



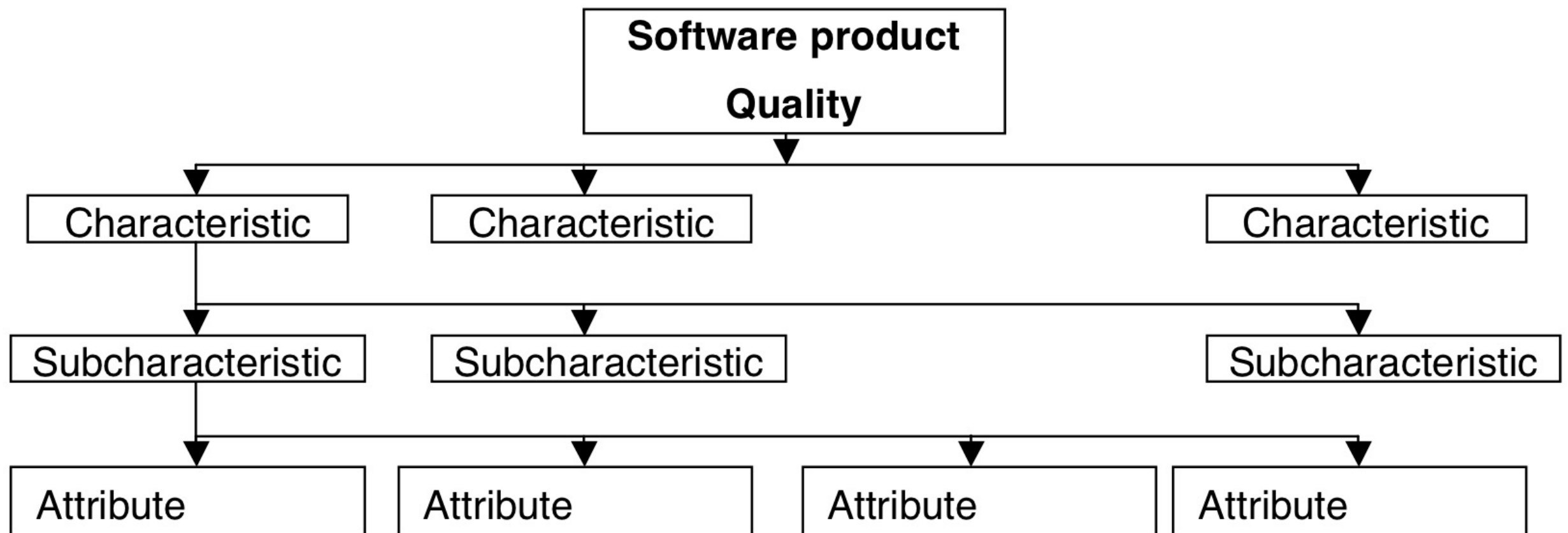
ISO/IEC 9126

- ❖ (Leváltotta az ISO/IEC 25010:2011)
- ❖ Szoftvertermék minőségi szabvány
- ❖ Projekt prioritások <- minőség modell : jellemzők (faktorkok) -> aljellemezők (kritériumok)
- ❖ A jellemzők (externálisan) manifesztálódnak a szoftver használatakor
 - ❖ a belső attribútumok következményeiként
- ❖ 6 faktor

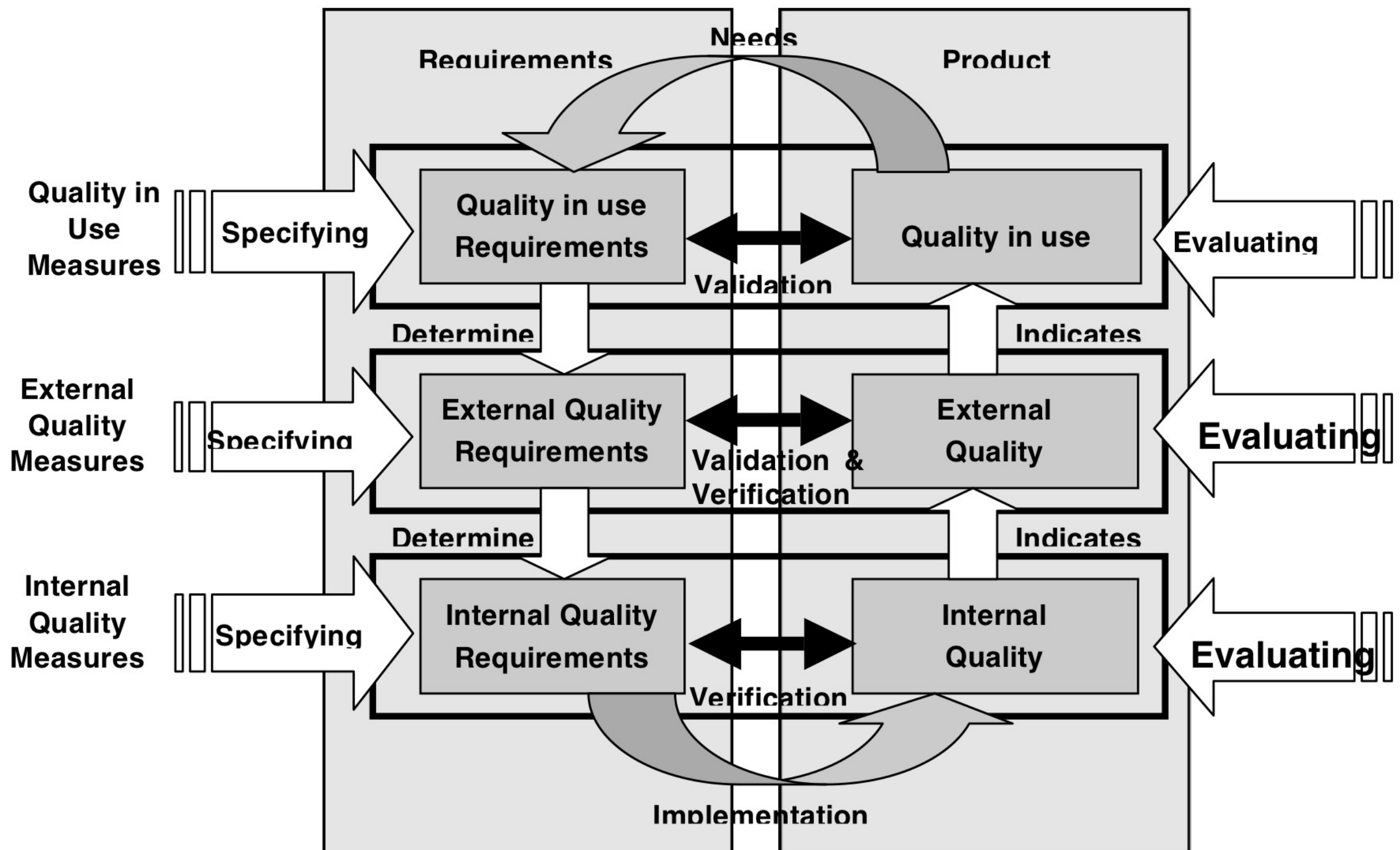
ISO/IEC 9126

- ❖ Funkcionalitás
 - ❖ alkalmasság
 - ❖ pontosság
 - ❖ együttműködtethetőség
 - ❖ biztonságosság (security)
 - ❖ megfelelés (szabványoknak)
- ❖ Megbízhatóság
 - ❖ érettség
 - ❖ hibatűrés
 - ❖ helyreállási képesség
- ❖ Használhatóság
 - ❖ megérthetőség
 - ❖ tanulhatóság
 - ❖ működtethetőség
 - ❖ vonzóság
- ❖ Újra felhasználhatóság
- ❖ Hatékonyság
 - ❖ időbeli viselkedés
 - ❖ erőforrás felhasználás
- ❖ Karbantarthatóság
 - ❖ analizálhatóság
 - ❖ változtathatóság
 - ❖ stabilitás
 - ❖ tesztelhetőség

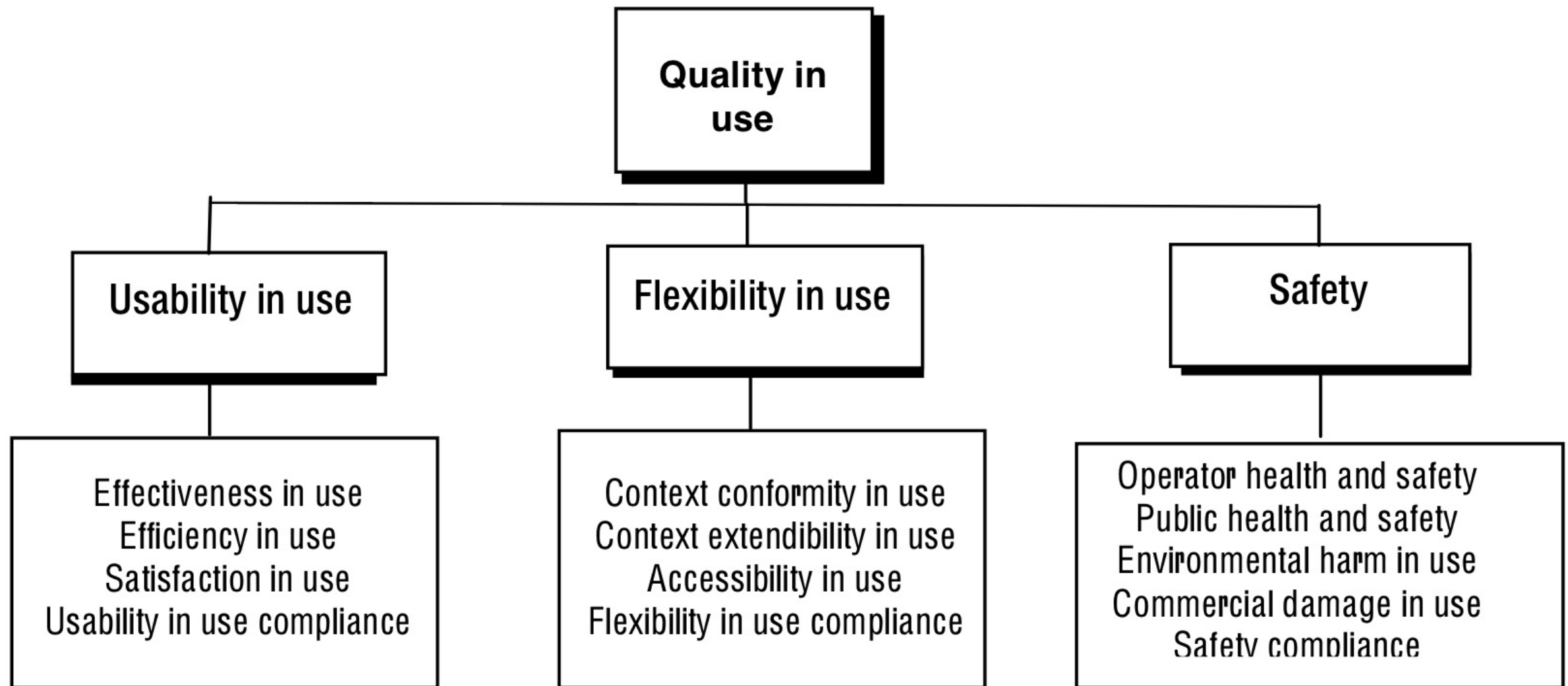
ISO/IEC 25010:2011



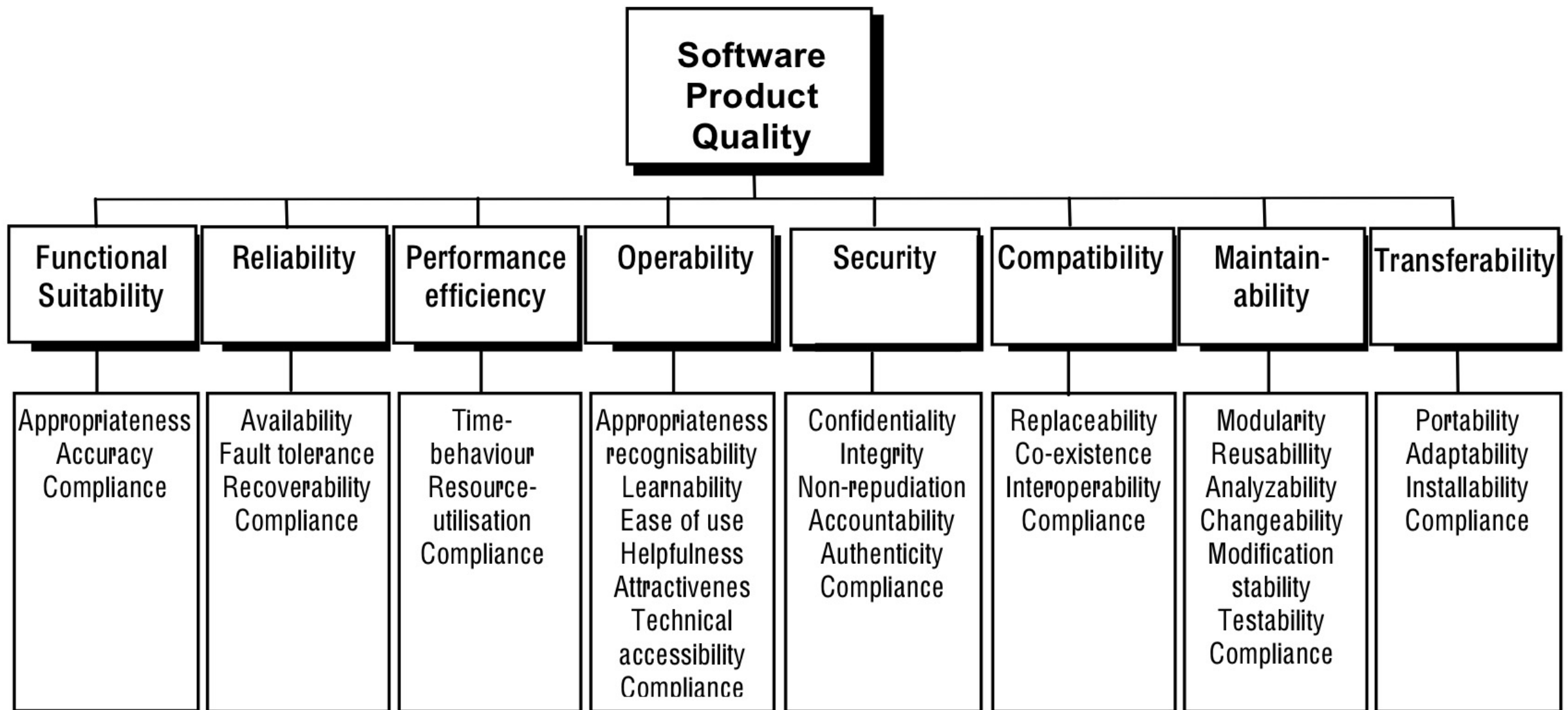
ISO/IEC 25010:2011



ISO/IEC 25010:2011



ISO/IEC 25010:2011



CENELEC 50128

- ❖ Vasúti alkalmazások - Kommunikáció, jelző és feldolgozó rendszerek
- Vasúti irányító és biztosító rendszerek szoftverei
- ❖ EN 50128 Szoftver
 - ❖ Szoftverek, szoftverek közötti kölcsönhatások
 - ❖ Alkalmazás programok
 - ❖ Operációs rendszerek
 - ❖ Támogató eszközök
 - ❖ Firmware
- ❖ További kapcsolódó szabványok
 - ❖ EN 50129 System safety
 - ❖ EN 50159 Communication
 - ❖ EN 50126 RAMS

CENELEC 50128 (folyt.)

- ❖ Alkalmazás:
 - ❖ retroaktívan létező rendszerekre / nincs retrospektív ellenőrzés
 - ❖ alkalmazható módosítások és kibővítés esetén
 - ❖ új fejlesztésekre kötelező
- ❖ Kiterjed:
 - ❖ szervezetekre: szerepek / kompetenciák
 - ❖ életciklus tevékenységekre, dokumentációra
 - ❖ V-modellt ír elő
- ❖ Forma:
 - ❖ Záradék táblázatok (clauses tables)
 - ❖ életciklus tevékenységek
 - ❖ Részletes táblázatok (detailed tables)
 - ❖ kódolási standard, funkcionális tesztelés, programozási nyelvek, modellezés

CENELEC 50128 (folyt.)

- ❖ A szabványban megfogalmazott követelmények:
 - ❖ Az életciklus tevékenységnél alkalmazandó technikák az egyes SIL szintekre
 - ❖ A részletes táblázatokban megadva:
 - ❖ **nem ajánlott, ajánlott, nagyon ajánlott, kötelező**
- ❖ pl. architektúra:

Technique	SIL 0	SIL 1	SIL 2	SIL 3	SIL 4
... Error detecting codes	-	R	R	HR	HR
... Backward Recovery	-	NR	NR	NR	NR
... Fully defined interfaces	HR	HR	HR	M	M

CENELEC 50128 (folyt.)

❖ Biztonság integritási szintek (SIL)

SAFETY INTEGRITY LEVEL	DEMAND MODE ("LOW DEMAND MODE")		CONTINUOUS MODE ("HIGH DEMAND MODE")	CONSEQUENCE OF A FAILURE
Level	AVAILABILITY	Probability of a failure on demand	Probability of a dangerous failure per hour	
SIL 4	>99.99%	$\geq 10^{-5}$ to 10^{-4}	$\geq 10^{-9}$ to 10^{-8}	Potential for fatalities in the community
SIL 3	99.9%	$\geq 10^{-4}$ to 10^{-3}	$\geq 10^{-8}$ to 10^{-7}	Potential for multiple fatalities
SIL 2	99%-99.9%	$\geq 10^{-3}$ to 10^{-2}	$\geq 10^{-7}$ to 10^{-6}	Potential for major injuries or one fatality
SIL 1	90-99%	$\geq 10^{-2}$ to 10^{-1}	$\geq 10^{-6}$ to 10^{-5}	Potential for minor injuries
SIL 0	Safety impact below SIL 1			

CENELEC 50128 (pl.)

Table A.5 – Verification and Testing

Technique / measure	Reference	SIL 0	SIL 1	SIL 2	SIL 3	SIL 4
Static Analysis	Table A.19	-	HR	HR	HR	HR
Perform Static Analysis using a customizable set of Coding Standards, Control Flow and Data Flow Analysis Rules. See Table A.19 for details.						
Dynamic Analysis and Testing	Table A.13	-	HR	HR	HR	HR
Perform Unit Test execution (on different levels: from file-scope to project-scope) and Application Monitoring – both with Code Coverage Analysis. See Table A.13 for details.						
Metrics		-	R	R	R	R
Use Code Metrics and Coding Standards modules to calculate a wide range of code metrics, including complexity metrics, coupling between objects, depth of inheritance, etc.						
Technique / measure	Reference	SIL 0	SIL 1	SIL 2	SIL 3	SIL 4
Traceability		R	HR	HR	M	M
Connect Unit Tests with requirements, development tasks or defects by inserting the appropriate information as metadata in code comments.						
Test Coverage for Code	Table A.21	R	HR	HR	HR	HR
Use the Coverage Module to report Code Coverage for the executed Unit Tests and for the functional tests executed by monitoring the running application. See Table A.21 for details.						
Functional/ Black-box Testing	Table A.14	HR	HR	HR	M	M
Execute the Unit Tests prepared to verify the functionality of the developed code. See Table A.14 for details.						
Interface Testing		HR	HR	HR	HR	HR
Execute the Unit Tests that were automatically generated using a wide range of input values for the tested functions (boundary values, characteristic parameter values, user-defined values for given types, etc.) Use external data sources to define a single Unit Test with a number of input combinations.						

CENELEC 50128 (folyt.)

- ❖ Eszközök osztályozása:
- ❖ T1 osztály
 - ❖ a kimenet nem eredményez a szoftverben végrehajtható kódot / felhasznált adatokat
- ❖ T2 osztály
 - ❖ a tesztelést / verifikációt érintő eszköz (hibás működés szoftverhibák felfedezését érinti)
- ❖ T3 osztály
 - ❖ a kimenet közvetlenül végrehajtásra kerülő kód / felhasznált adat