

Szoftver verzió- és konfigurációmenedzselés

Bevezetés

A világ szoftveriparának történelme, de sajnos jelene is, tele van sikertelen projektekkel. Ezek számának csökkentése csak megfelelő módszerek fegyelmezett betartásával lehetséges.

Az Amerikai Egyesült Államok védelmi minisztériuma által, a Carnegie-Melon University-nél az 1980-as évek közepén, létrehozott Software Engineering Institute (SEI) feladata a védelmi minisztériumnak fejlesztő szoftver vállalatok képességének növelése volt. Ennek a képességfelmérő munkának az eredménye az SEI Software Capability Maturity Model (CMM), képesség fejlettségi modell, lett, mely öt lehetséges fejlettségi szintjét határozta meg a vállalatoknak. Az első verzió után, mely azt a kritikát kapta, hogy nem volt kellően precíz, 1993-ban megjelent a második verzió [8], amely megtartotta az öt szintet, de ezeket kifejezettebben, kulcsfolyamat területekkel határozta meg. A CMM és a nemzetközi minőségbiztosítási szabvány, az ISO 9000 között van egy nyilvánvaló korreláció, de a CMM részletesebb és normatívabb és tartalmaz egy keretet az eljárás javítására [10, p. 571].

A fejlettség második szintje, az úgynevezett „repeatable process”, a megismételhető folyamat szintje (melynek elérése általában biztosítja az ISO 9000 minőségtanúsítványt [10, p. 571]) a következő kulcsfolyamat területeket tartalmazza:

Requirements management, követelmények menedzselése

Software project planning, projekttervezés

Software projekt tracking and oversight, projekt követés és áttekintés

Software subcontract management, alvállalkozói megállapodások menedzselése

Software configuration management, konfigurációmenedzselés

Software quality assurance, minőségbiztosítás

Ezek közül korábban megvizsgáltuk a minőségbiztosítást [1], most, a konfigurációmenedzselést [6, 7, 11] vizsgáljuk meg közelebbről.

Konfigurációmenedzselés

Minden szoftverprojekt tervének tartalmaznia kell a projekt konfigurációmenedzselési tervét [9, p.118]. A projekttervhez tartozó minőségbiztosítási terv [1, 3] külön is előírja a konfigurációmenedzselési terv készítését és annak folyamatos felülvizsgálatát.

A következőkben, a legfontosabb definíciók megadása után, röviden ismertetjük a konfigurációmenedzselést, a konfigurációmenedzselési rendszert, majd külön fejezetben, a konfigurációmenedzselési terv IEEE – szabványának [4] szerkezetét, mely megfelel az ISO – szabvány előírásainak [5].

Legfontosabb definíciók a [2] szerint:

Konfiguráció (configuration): Hardvernek vagy szoftvernek technikai dokumentációban kifejtett vagy egy termékben megvalósított funkcionális és fizikai karakterisztikái.

Konfigurációtétel (configuration item): Konfigurációmenedzseléshez kiválasztott és a konfigurációmenedzselési folyamatban egyedi entitásként kezelt hardver vagy szoftver, vagy mindkettő aggregációja.

Verzió (version): Egy kezdeti kiadása vagy újrakiadása egy szoftver konfigurációtételnek, társítva a teljes fordításával vagy újrafordításával a kód esetében.

Alapverzió (baseline): Egy formálisan felülvizsgált és megegyezés szerinti specifikáció vagy termék, amely ezután a további fejlesztés alapjaként szolgál és amelyet, csak formális változtatásellenőrző procedúrán keresztül lehet megváltoztatni.

Felépítés (build): Egy működő verziója egy rendszernek vagy komponensnek, mely magába foglalja egy megadott részhalmazát a végleges termék funkcióinak.

Mi is a konfigurációmenedzselés? Röviden kifejezve, a konfigurációmenedzselés irányítja a változtatásait és fenntartja az egységességét egy projekt összes létrehozott tárgyi elemének. A konfigurációmenedzseléshez tartozik a

- Konfigurációtételeknek az azonosítása,
- Ezen tételek változtatásának korlátozása
- Ezen tételek változtatásának ellenőrzése
- Ezen tételekből összeállított konfigurációk meghatározása és menedzselése

A metódusokat, a folyamatok és az eszközök melyek lehetővé teszik a változtatásokat és a konfigurációmenedzselést egy vállalaton belül, együttesen, konfigurációmenedzselő rendszernek nevezzük.

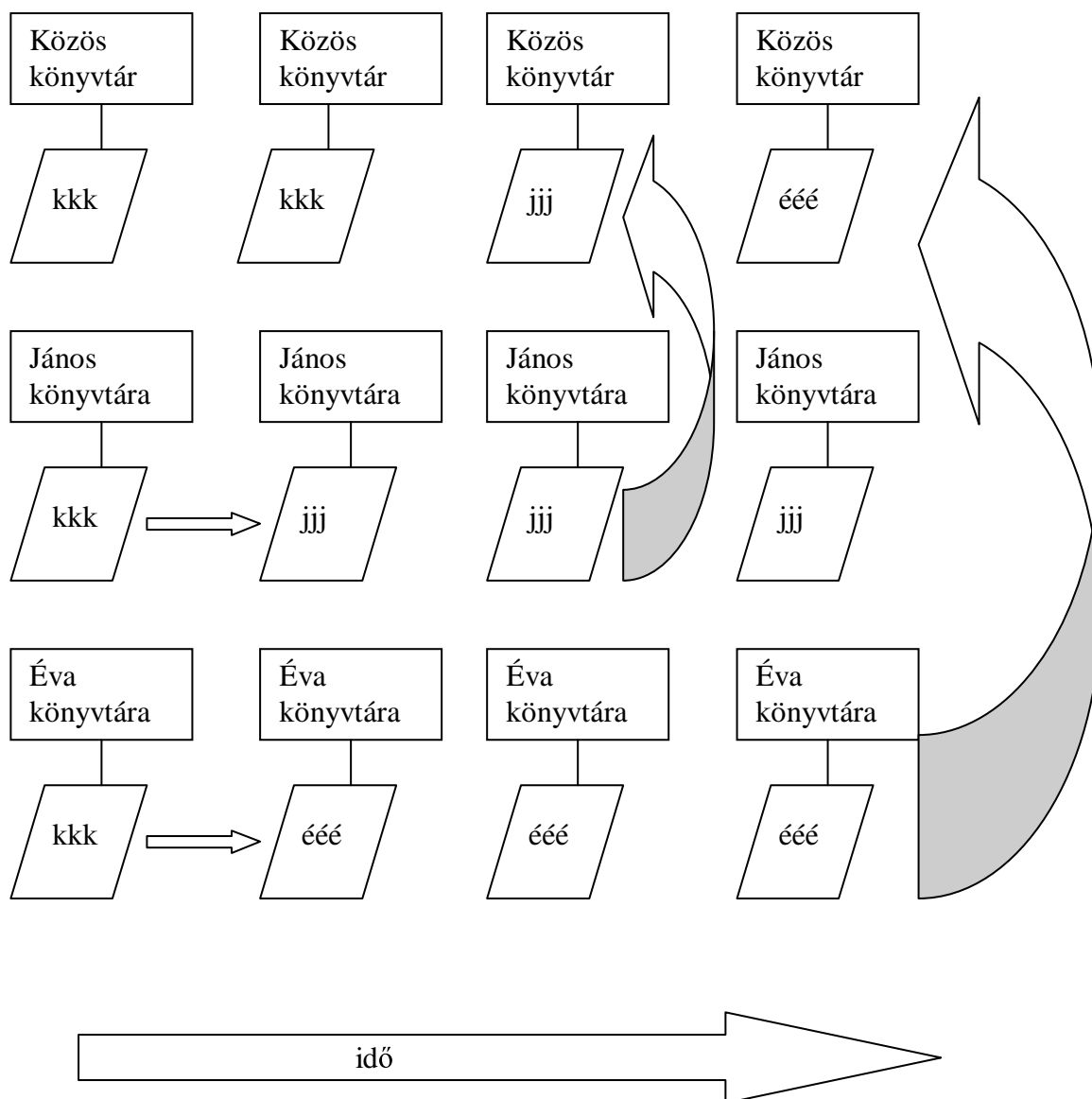
A konfigurációmenedzselő rendszer formálisan a konfigurációmenedzselési tervben van meghatározva, melynek a [4] alatt megadott szabvány szerinti vázlatát a következő fejezetben ismertetjük. Most a konfigurációmenedzselő rendszer célját, majd az alapverziók szerepét ismertetjük.

A konfigurációmenedzselő rendszer célja

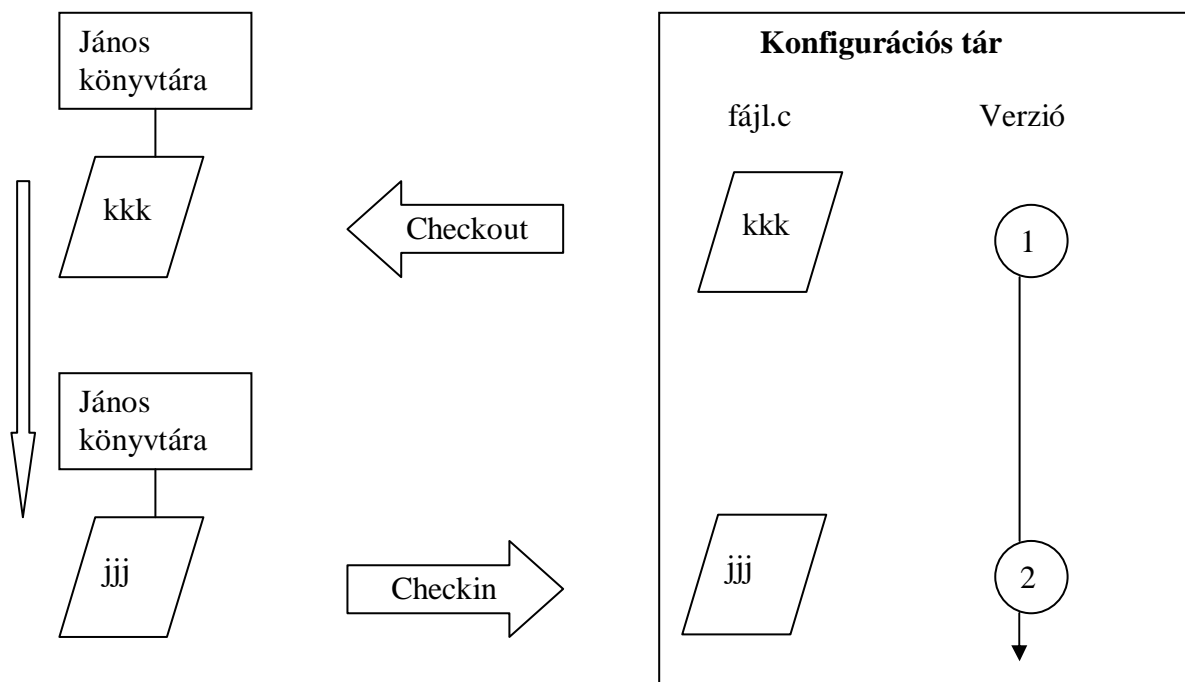
A konfigurációmenedzselő rendszer elsődleges fontosságú szerepet tölt be egy közös projekten dolgozó munkatársak által készített számos tárgyi elem létrejöttének szabályozásában és felügyeletében. A szabályozás segít a költséges félreértések elkerülésében és biztosítja azt, hogy a létrehozott tárgyi elemek ne legyenek konfliktusban egymással a következő típusú problémák miatt:

- Egyidejű frissítés
- Korlátozott közlés
- Többszörös verziók

Egyidejű frissítés Amikor kettő vagy több munkatárs dolgozik ugyanazon a tárgyi elemen, akkor az, aki utoljára változtat, megsemmisíti az előzők munkáját. A probléma az, hogyha a rendszer nem támogat egyidejű frissítést, akkor a változtatások csak sorozatosan történhetnek, ami lelassíthatja a fejlesztési folyamatot. Ha a rendszer támogatja az egyidejű frissítést, akkor azt fel kell tudni ismernie és az egyidejűleg frissített tárgyi elemek közötti különbségeket integrálnia kell, mielőtt a változtatások mentésre kerülnének.



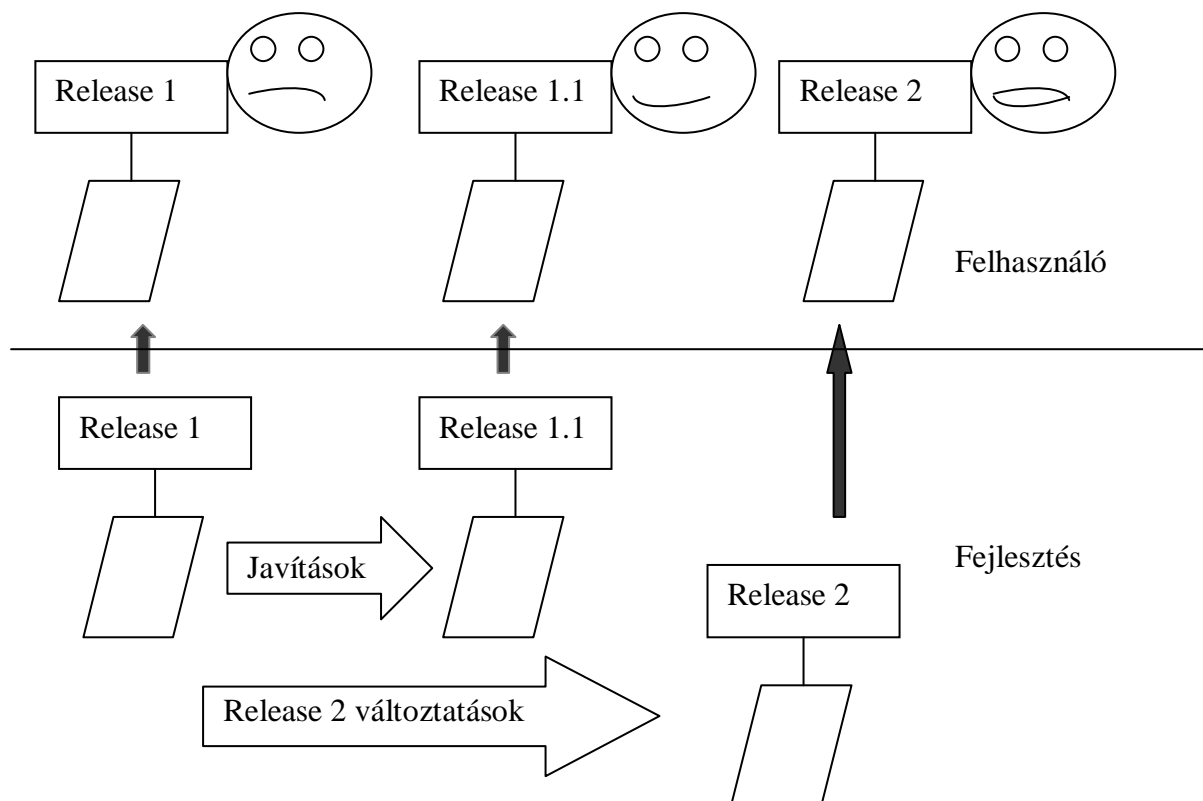
Ábra 1 – Az egyidejű frissítés problémája



Ábra 2 – A checkout és ckeckin műveletek

Korlátozott közlés Amikor egy tárgyi elemben lévő hiba kijavításra kerül és az nem lesz tudomására hozva minden munkatársnak aki a tárgyi elemen dolgozik.

Többszörös verziók A legtöbb nagy program evolúciós kiadásokban van fejlesztve. Egy kiadás fogyasztói felhasználás alatt lehet, egy másik tesztelés alatt, míg egy harmadik fejlesztés alatt. Ha hiba van jelentve akármelyik verzióban, akkor a kijavítást ki kell terjeszteni minden verzióra. Itt félreértések léphetnek fel, melyek költséges kijavításokhoz és újra elvégzett munkához vezethetnek akkor, ha a változtatások nincsenek gondosan szabályozva és felügyelve.



Ábra 3 – Hibák újra bevezetése két kiadás között

Egy konfigurációmenedzselő rendszer felhasználható fejlesztett szoftverrendszerek többszörös variánsainak menedzselésére, annak nyomon követésére, hogy melyik verziók vannak felhasználva egy adott felépítésben, egyedi programokból vagy teljes kiadásokból álló felépítések létrehozására felhasználó által definiált verzióspecifikációk szerint, és vállalat specifikus fejlesztési irányvonalak érvényre juttatására.

Néhány közvetlen előny melyet egy konfigurációmenedzselő rendszer nyújt:

- Fejlesztési folyamatok támogatása
- A termék integritásának (sértetlenség, egység) megőrzése
- Biztosítja a teljességet és helyességet a konfigurált terméknek
- Gondoskodik egy stabil környezetről a termék fejlesztéséhez
- A projektirányvonalaknak megfelelően korlátozza a tárgyi elemek változtatását
- Gondoskodik egy ellenőrzési nyomról azt illetően, hogy miért, mikor és ki változtatta meg akármelyik tárgyi elemet.

Továbbá, egy konfigurációmenedzselő rendszer részletes nyilvántartási adatokat tárol a fejlesztési folyamatról magáról: ki hozott létre egy bizonyos verziót (és miért és mikor), milyen verziók kerültek bele egy bizonyos építésbe, és más lényeges információkat.

Az alapverziók szerepe

Egy alapverzió egy, a projekt adatbázisában lévő összes tárgyi elem egy verziójából álló, pillanatkép. Ez hivatalos mértékül szolgál, amelyre a további munkát alapozni lehet, és amelyet csak formális változtatásellenőrző procedúrán keresztül lehet megváltoztatni. Miután egy első alapverzió létrejött, minden következő változtatás delta-ként van nyilvántartva a következő alapverzió létrejöttéig.

Ha pl. a projekthez egy új munkatárs csatlakozik, feltöltheti a munkaterületét az aktuális alapverzióval, és így szinkronba kerül a projekten dolgozó többi munkatárssal. Amint a fejlesztés továbbhalad, a fejlesztők változtatásokat (deltákat) eszközölnek az aktuális verzión, amelyekkel a többi munkatársak rendszeresen kiegészíthetik a munkaterületükben lévő verziót. Ezt újraalapozásnak (rebasin - nek) hívjuk, mely művelet egybeolvasztja a munkaterületben lévő fájlokat a projekt adatbázisban lévő legaktuálisabb fájlokkal.

A három fő oka az alapverziók létrehozásának a reprodukálhatóság, a nyomon követhetőség és a jelentés nyilvántartás.

A **reprodukálhatóság** az a képesség mely lehetővé teszi azt, hogy időben visszamenve létre lehessen hozni egy adott kiadását egy szoftver rendszernek, vagy reprodukálni egy fejlesztési környezetet a projekt korábbi idejéből. A **nyomon követhetőség** létrehozza az előd–utód kapcsolatot a projekt tárgyi elemei között. Ennek a célja az, hogy biztosítani lehessen hogy a design megfelel a követelményeknek, a kód implementálja designt, és hogy a futtatható rendszer a helyes kódból van építve. A **jelentés nyilvántartás** az egyik alapverziónak a másikkal történő összehasonlításán alapul.

Mikor egy új alapverzió létre lesz hozva, akkor annak minden alkotó komponensét és minden alapverziót egyértelműen kell jelölni, azért hogy azok azonosíthatóak és újra létrehozhatóak legyenek.

Több előnye van alapverziók létrehozásának:

- Egy alapverzió egy stabil pontot és egy pillanatképet nyújt a fejlesztett tárgyi elemekről.
- Alapverziók egy stabil pontot nyújtanak, ahonnan új projekteket lehet létrehozni. Az új projektet, mint egy szeparált ágat, az eredeti projekt további változtatásaitól elszeparálva kell létrehozni.
- Egyéni fejlesztők egy alapverzió komponenseit a privát munkaterületükben lévő komponensek frissítésének alapjául vehetik.
- Egy alapverzió lehetővé teszi egy munkacsoport számára, hogy változtatásokat vizsgálgyördítsen, ha azok labilisok vagy kétesek.
- Egy alapverzió lehetővé teszi jelentett hibák reprodukálását, mivel az a konfiguráció újra létrehozható, mely az adott kiadást képezte.

Milyen gyakran hozzunk létre alapverziókat? A válasz az, hogy gyakran, hogy a fejlesztők szinkronban legyenek egymás munkájával. Akárhogy is, egy projekt folyamán alapverziókat rutinszerűen kell létrehozni minden iteráció végén (kis mérföldkövek) és a nagy mérföldköveknél, melyek a az életciklus különböző fázisainak végét jelölik.

Az IEEE Std 828-1998, Konfigurációmenedzselési terv szabványának szerkezete

A szabvány előírja, hogy a konfigurációmenedzselési terv hat fejezetből álljon hat különböző információosztálynak megfelelően az alábbiak szerint.

1. Bevezetés

A terv célja

A terv alkalmazásának hatálya

A szoftver projekt összefoglaló leírása

Azon szoftver konfigurációtételek azonosítása, melyekre a konfigurációmenedzselést alkalmazni kell

Más szoftver azonosítása, amelyet a tervbe be kell venni (pl. szupport vagy teszt szoftver)

A projekt konfigurációmenedzselésének kapcsolata a hardver vagy az azon lévő rendszer konfigurációmenedzseléséhez

A formalitás foka, a felügyelet mélysége, és a szoftver életciklusának azon részei, melyekre a konfigurációmenedzselést alkalmazni kell a projekten belül

Korlátozások, pl. a tervre érvényes időkorlátozások

Feltételezések, melyek kihatással lehetnek a költségekre, az ütemezésre, vagy a meghatározott konfigurációmenedzselési tevékenységekre (pl. feltételezések a megrendelőnek a konfigurációmenedzselési tevékenységekben való részvételének fokáról, vagy automatizált segédeszközök hozzáférhetőségéről)

A terv kulcsszavai

A terv utalásai

2. Konfigurációmenedzselés irányítása

(Ki?) Azonosítja a felelőségeket és a tervezett tevékenység elvégzéséért felelős szerveket.

Szervezés

Szervezési egységek, melyek részt vesznek vagy felelősek bármilyen konfigurációmenedzselési tevékenységért a projekten belül

Ezen szervezési egységek funkcionális szerepe a projekt szerkezetén belül

A kapcsolat a szervezeti egységek között

Konfigurációmenedzselési felelősségek

Szándék és célkitűzések

Tagság és elkötelezettség

A hatályban lévő időszak

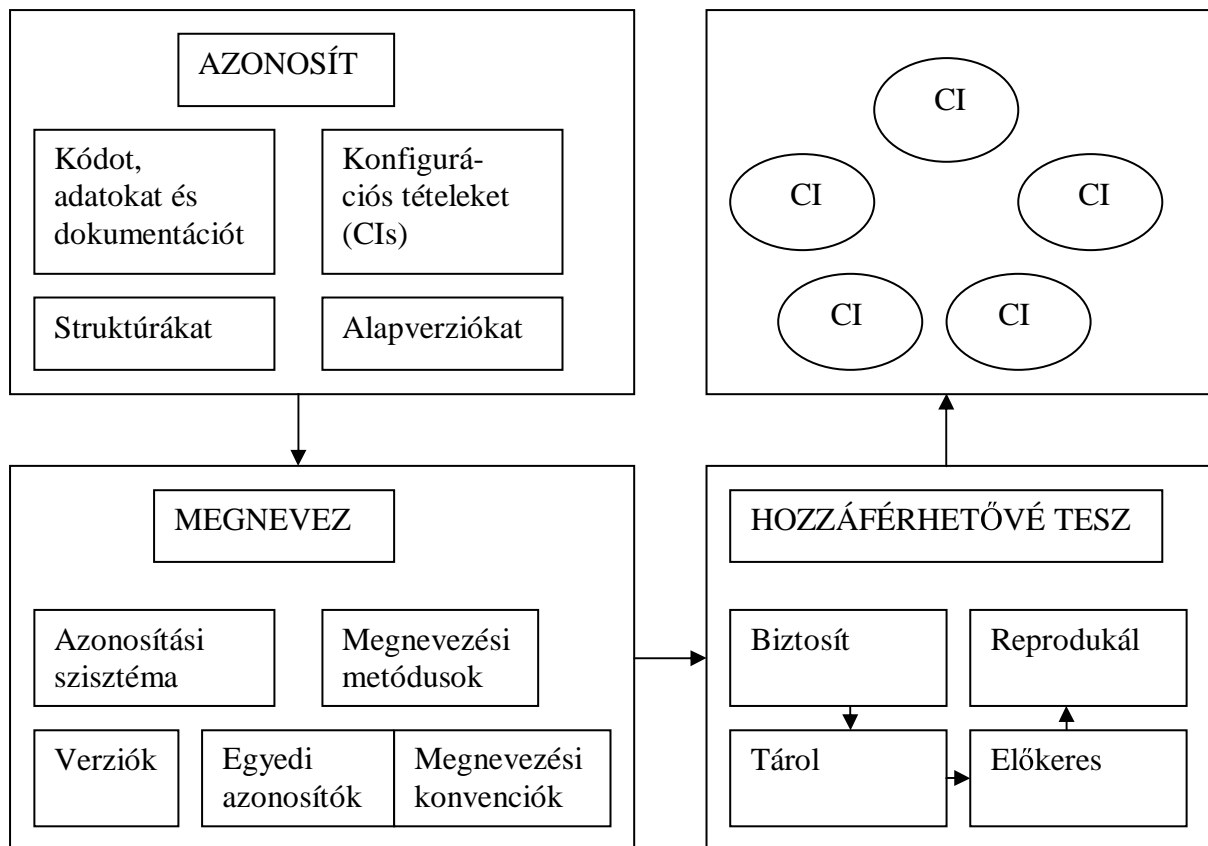
A hatáskör területe

Operatív procedúrák

Alkalmazható irányvonalak, direktívák és procedúrák

3. Konfigurációmenedzselési tevékenységek

(Mit?) Azonosít minden, a projekten alkalmazandó elvégzendő tevékenységet.



Ábra 4 – Konfigurációazonosító folyamatok

Konfiguráció azonosítás

Konfigurációtételek azonosítása

Az esemény amely egy alapverziót hoz létre

A tételek melyeket ellenőrizni kell az

- alapverzióban
- Az alkalmazandó procedúrák az alapverzió létrehozására és változtatására
- A szükséges hatáskör a jóváhagyott alapverzió dokumentációja megváltoztatásának jóváhagyásához
- Konfigurációtételek megnevezése
- Konfigurációtételek hozzáférhetősége
- Konfiguráció irányítás
 - Változtatások megkérése
 - A neve és verziója a konfigurációtételnek, amelyben a probléma felmerül
 - A megkérő neve és vállalata
 - A megkérés dátuma
 - Sürgősségi indikáció
 - A változtatás szükségessége
 - A megkért változtatás leírása
 - Változtatások értékelése
 - Változtatások jóváhagyása vagy jóvá nem hagyása
 - Változtatások implementálása
 - A kapcsolatban lévő változtatás megkérések
 - A nevei és verziói az érintett tételeknek
 - A hitelesség dátuma és a felelős fél
 - Kiadás vagy installáció dátuma és a felelős fél
 - Az új verzió azonosítója
- Konfiguráció státuszának nyilvántartása
 - Milyen adatokat kell nyomon követni és jelenteni az alapverzióknál és a változtatásoknál
 - Milyen típusú státusznyilvántartó jelentéseket kell generálni és milyen gyakran
 - Hogyan kell az információt összegyűjteni, tárolni, feldolgozni és jelenteni
 - Hogyan kell a státuszadatokhoz való hozzáférhetőséget ellenőrizni
- Konfiguráció ellenőrzések és felülvizsgálatok (audits and reviews)
 - A tervnek minden tervezett konfiguráció ellenőrzéshez vagy felülvizsgálathoz definiálnia kell a következőket:
 - Annak célkitűzését
 - Az ellenőrzés vagy felülvizsgálat alatt lévő konfigurációtételeket

Az ellenőrzési vagy felügyeleti feladatok munkatervét

Az ellenőrzés vagy felügyelet lebonyolításának procedúráját

A résztvevőket munkahelyi beosztásuk megjelölésével

A dokumentáció, melynek hozzáférhetősége szükséges az ellenőrzéshez vagy felügyelethez, vagy annak támogatásához

A procedura minden hiányosság nyilvántartására és a javító intézkedések jelentésére

A jóváhagyási kritérium és a jóváhagyást követő speciális tevékenységek

Interfész felügyelet

A tervnek azonosítani kell a külső tételeket, melyekhez a projektnek van csatlakozó felülete (interfésze).

Minden interfészhez a tervnek definiálnia kell a következőket:

A jellegét az interfésznek

Az érintett vállalatot

Hogyan legyen felügyelve az interfész kódja, dokumentációi és adatai

Hogyan legyen az interfész felügyelet dokumentációja jóváhagyva és rendelkezésre bocsátva egy adott alapverzióhoz

Alvállalkozó/szállító felügyelet

Alvállalkozói szoftverre vonatkozóan a tervnek a következőket kell előírnia:

Milyen konfigurációmenedzselési

követelmények, beleértve a konfigurációmenedzselési tervet, legyenek részei az alvállalkozói megállapodásnak

Hogyan legyen felügyelve az alvállalkozó a követelmények betartását illetően

Milyen konfiguráció ellenőrzései és felülvizsgálatai legyenek megtartva az alvállalkozói tételeknek

Hogy legyen tesztelve, hitelesítve, elfogadva és egyesítve a projektszoftverrel a külső kód, a külső dokumentáció, és a külső adatok

Hogyan legyenek menedzselve magántulajdonban (pl. szerzői jog és százalékos honorárium) lévő tételek az

információ megvéde és a tulajdonjog
követhetősége érdekében
Hogyan legyenek a változtatások végrehajtva,
beleértve az alvállalkozó részvételét

Beszerzett szoftverre vonatkozóan a tervnek elő kell
írnia, hogy hogyan legyen a szoftver átvéve, tesztelve
és konfigurációmenedzselés alá helyezve.

4. Konfigurációmenedzselés munkaterve

(Mikor?) Azonosítja a szükséges koordinációt a
konfigurációmenedzselési tevékenységek és a projekt többi
tevékenysége között.

5. Konfigurációmenedzselés forrásai

(Hogyan?) Azonosítja a terv végrehajtásához szükséges eszközöket,
és fizikai és emberi erőforrásokat

6. Konfigurációmenedzselési terv karbantartása

Azonosítja azt, hogy hogyan lehet a tervet érvényben tartani, míg
hatályban van

Ki a felelős a konfigurációmenedzselési terv karbantartásáért
Milyen gyakran kell frissítéseket végrehajtani
Hogyan kell a terv megváltoztatását kiértékelni és jóváhagyni
Hogyan kell a terv megváltoztatását végrehajtani és közzé
tenni

A szabvány külön fejezetekben tárgyalja a tervnek a projekt körülményeihez való illesztését
és a szabványnak való megfeleltetését.

A konfigurációmenedzselési terv alkalmazásához az útmutatót az IEEE Std 1042-1987
szabvány adja meg.

Irodalomjegyzék

1. F. Belik: „Szoftver minőségbiztosítás”, ELTE, Ászt, 2002.
2. IEEE Std 610.12-1990: „*IEEE Standard Glossary of Software Engineering Terminology*”, <http://www.ieee.org>.
3. IEEE Std 730-1998 (Revision of IEEE Std 730-1989): „*IEEE Standard for Software Quality Assurance Plans*”, <http://www.ieee.org>.
4. IEEE Std 828-1998: „*IEEE Standard for Software Configuration Management Plans*”, <http://www.ieee.org>

5. ISO 10007, First edition 1995-04-15: „*Quality management – Guidelines for configuration management*”, <http://www.iso.org>.
6. A. Leon: „*A Guide to Software Configuratin Managment*”, Artech Haus, 2000.
7. T. Mikkelsen, S. Pherigo: „*Practical Software Configuration Management*”, Prentice Hall, 1997.
8. M. Paulk, C. V. Weber, B. Curtis, M. B. Chrissis: „*The Capability Maturiti Model: Guidelines for improving the Software Process*”, Addison-Wesley, 1994.
9. S. L. Pfleeger: „*Software Engineering, Theory and Practice*”, 2nd Edition, Prentice Hall, 2001.
10. I. Sommerville: „*Software Engineering*”, 6th Edition, Addison-Wesley, 2001.
11. B. A. White: „*Software Configuration Strategies and Rational ClearCase*”, Addison-Wesley, 2000.