

6.8.

Szoftveripar sajátosságai Dr. Balla Katalin

6.8.1. Bevezető. A szoftvert sajátossá tevő elemek

Manapság mindennapossá vált a „szoftver” emlegetése. Míg korábban az embereknek csak kis hányada, jóformán „laboratóriumi körülmények között” foglalkozott szoftverrel, addig ma már általánossá vált, hogy a nagy áruházak polcairól leemeljük és bevásárlókosarunkba tesszük az éppen szükséges „programot”. Szoftver „vezeti” a repülőgépeket, szoftver „navigálja” a taxisofőröket, szoftver ügyel helyettünk a gépkocsink állandó sebességére, az orvos keze helyett szoftver vezérli a műtéteket, szoftver mozgatja a pénzünket a bankban, szoftvert veszünk igénybe, ha gyerekünk felvételi eredményét meg akarjuk tudni – és a példákat még jócskán sorolhatnánk.

Életünk minden területébe beleszól a „szoftver”: segít, szórakoztat, vagy éppen bosszúságot okoz. Egy azonban biztos: befolyásolja életünket, s emiatt nem közömbös, milyen is a szoftver. Ismerőseinktől érdeklődünk, mikor szoftvert keresünk: vajon melyik a „jó”?

Valószínűleg kevesen kerülünk olyan helyzetbe, amikor valaki visszakérdez: pontosan milyen a „jó” szoftver? Mit válaszolnánk vajon erre a kérdésre...?

A válasz biztosan nem jönne könnyen, hiszen „attól függ”, hogy éppen milyen szoftvert keresünk, mennyi a pénzünk, milyen az otthoni számítógépünk kapacitása, milyen operációs rendszer fut rajta, dolgoztunk-e már korábban is szoftverrel, tudunk-e angolul, vagy csak a magyar nyelvű programot tudnánk elindítani ...Ilyen problémákkal nemcsak mi, egyéenként, hanem azok a cégek is küzdenek, amelyek nem kész szoftvert, hanem „rendelésre, méretre” gyártottat kívánnak alkalmazni. Fontos lehet nekik, hogy a szoftver időben elkészüljön, kódja könnyen érthető és könnyen karbantartható legyen, sokfajta környezetben legyen telepíthető, biztonságosan kezelje a cég és ügyfeleinek adatait, akkor is válaszoljon belátható időn belül, amikor több tízezren futtatják egy időben...

Nem könnyű tehát a szoftver „jóságát”, minőségének kritériumait megfogalmazni. Érdekes talán megvizsgálni, mitől ez a nehézség? Arról van szó ugyanis, hogy egy termék minőségének jellemzőit szeretnénk meghatározni, és ez látszólag eléggé egyértelmű feladat. Ha azonban bútort vásárolunk / rendelünk, kevesebb nehézséggel küzdünk, hiszen van elképzelésünk arról, hogy konyhabútort vagy nappali szobába való alkalmatosságot keresünk, tudjuk, mekkora helyen kell elférnie, a színek között a kiállított minták alapján válogathatunk, és az árat is egészen pontosan megmondja az eladó vagy a bútorkészítő. Miben más vajon a szoftver?

Hát, először is, a szoftver igen sok elemből áll. Az ISO / DIS 9000:2000 szabvány definíciója szerint a „szoftver: szellemi termék, amely egy hordozó médiumon levő információkból áll.” Megjegyzésként hozzátesszik még: „A szoftver megjelenhet

koncepciók, ügyletek vagy eljárások alakjában. Egy példa a szoftverre a számítógépprogram.”

Ne tévesszük tehát össze a szoftvert a számítógépes programmal! A „programon” kívül (melynek fizikai valója – nehezen ugyan, de megfogalmazható, mondjuk: a számítógépen tárolt, futtatható file-ok és a működésükhöz szükséges adat-file-ok halmaza) tartalmaz ugyanis egy sor fizikai valójában nehezen tetten érhető elemet: az üzleti folyamatban alkalmazott fogalmakat, a lépéseket logikai sorrendbe állító tudást, a működtetést segítő szövegek érthetőségét...

Másik dolog, amely a szoftvert – ha nem is egyedivé, de – sajátossá teszi az, hogy igen gyorsan változnia kell, ha használatban kívánják tartani. Alkalmazkodnia kell a mindig megújuló hardver- és szoftverköznyezethez, amelyben működik, a technikai lehetőségek és életvitelbeli változások miatt olyan funkciókat kell ellátnia, amelyeket eredetileg nem terveztek bele (pl. kapcsolódás olyan eszközhöz, amely még nem is létezett, amikor a szoftvert fejlesztették).

A szoftverek nagy része célzottan, megrendelői igények alapján készül. A felhasználóknak viszont nehézséget okoz az igények pontos megfogalmazása, különösen akkor, ha korábban nem dolgoztak szoftverrel, vagy hirtelen egy, az eddigtől mindenben különböző technológiához kell adaptálniuk egy létező rendszert. Az igények specifikálásának nehézségeit az is fokozza, hogy a felhasználóknak irreálisan magas igényeik lehetnek egy szoftverrel szemben (azt várnák, például, hogy egy szoftver bevezetése oldja meg a vállalat irányítási, pénzügyi problémáit...)

Az előbb felsoroltakat figyelembe véve, tehetünk néhány megállapítást a szoftver minőségével kapcsolatban:

- A *szoftverminőség fogalma* nehezen meghatározható, mert – a dologi természetű javakkal ellentétben – nehéz a minőséget meghatározó kritériumokat pontosan megfogalmazni, a szoftvertermékek minőségét adekváтан mérő módszereket és eszközöket kidolgozni.
- A szoftver minősége nem „egy és egyetlen”, nem lehet általában megmondani, hogy mikor „jó” a szoftver.
- A szoftver minősége helyzetfüggő: a vevő igényei, a szoftver típusa, célköznyezete, a gyártó prioritásai mind befolyásolják a szoftver minőségének megítélését.
- A szoftver minősége sok elemtől függ, melyek között ajánlatos fontossági sorrendet kialakítani.
- Egyetlen szoftver minősége is mást és mást jelenthet különböző időpillanatokban, hiszen a szoftver működési köznyezete és maga a szoftver is igen gyorsan változik.

Ezekből a megállapításokból megérthető: a legtöbb, amit a szoftver minőségével kapcsolatban tehetünk, a *helyzetfüggő minőségi profil kialakítása*, majd a fontosnak ítélt minőségi attribútumok megfelelő értékének megvalósítása.

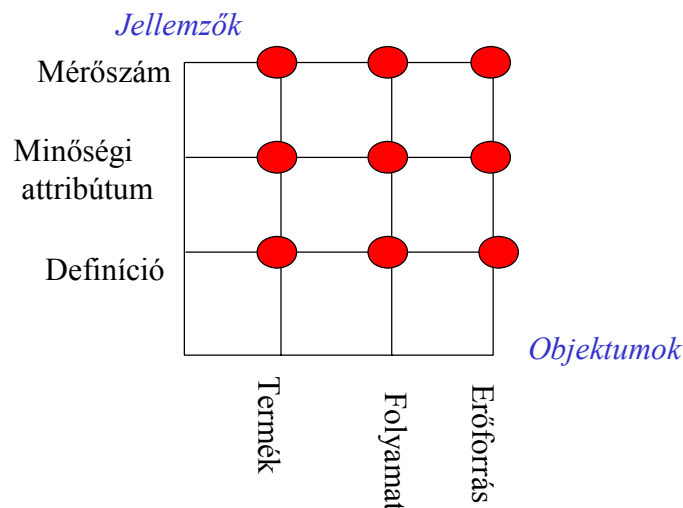
6.8.2. A szoftver minőségét befolyásoló elemek

A bevezetőben bemutattuk, a szoftver(gyártás) minőségében sokféle elem játszik szerepet, s a szoftver minőségének meghatározásához tudnunk kell, mi az elsődleges az

egyres esetekben. A tájékozódást segíti, ha végiggondoljuk, melyek azok az elemek, amelyek mindig jelen vannak a szoftver fejlesztésében, működtetésében. .

Minőségi szoftvergyártásához egyformán figyelniünk kell a szoftvertermékre, a szoftvert előállító folyamatra, az előállításban szerepet játszó erőforrásokra¹, és a folyamatokon belül kiemelten is a szoftvergyártási projektek irányítására. A szoftver minőségének értelmezésekor meg kell érteniünk a felsoroltakat, definiálniuk kell őket, majd ha bármelyik elem minőségét meg szeretnénk határozni, vagy figyelni akarjuk az alakulását, minőségi attribútumokat kell hozzájuk rendelni, s ezen attribútumok elvárásainknak való megfeleléséről mérésel is meg kell győződnünk.

Az eddigiek alapján felállítható a 6.8.1. ábrán látható minőségügyi keret. Ezt a QMIM-nek (Quality through Managed Improvement and Measurement) nevezett keretet részletesen leírja a [2] és a [3] sz. szakirodalmi forrás, ezért itt csak röviden ismertetjük.



6.8.1. ábra: A minőségügyi keret

Ha a szoftverminőség kérdésével a maga komplexitásában szeretnénk foglalkozni, nem elegendő egyetlen elemet kiemelni, hanem a 6.8.1. ábrán látható minőségügyi keretet teljes egészében ki kell tölteni. A kitöltés „sorrendjére”, „mélységére”, egyszóval: a minőségügyi keret alkalmazására nincs egyértelmű recept. A konkrét helyzet, a cég üzleti céljainak, lehetőségeinek, prioritásainak függvényében ki kell választani azokat az elemeket, amelyek az adott szituációban a legfontosabbak. Lényeges, hogy ne tévesszük szem elől a minőségügyi keret alapelvét: minden elem fontos, előbb vagy utóbb tehát mindegyikkel foglalkozni kell, ha a szoftverminőséget a maga teljességében értelmezzük. A minőségügyi keret kitöltése „vízszintesen” és „függőlegesen” egyaránt történhet. A „vízszintes kezdet” mellett dönteni azt jelenti, hogy a szoftvergyártás összes elemét megpróbáljuk megérteni, definiálni, és akkor lépünk a minőségi attribútumok szintjére,

¹ Ezek az elemek Fenton elméletében is szerepelnek (lásd, pl. [1]).

ha már minden elemet (és egymáshoz való viszonyukat) tökéletesen értünk. A keret „függőleges kitöltése” azt jelenti, hogy kiválasztunk egy elemet a vízszintes tengelyen, és annak minden vonatkozását tökéletesen megértjük, mielőtt egy másik elemmel foglalkozni kezdenénk.

A megközelítési módok sorrendjét nem kell szigorúan betartani. Lényeges, hogy az elemeket megértsük, a saját helyzetünkre alkalmazzuk őket, és fontossági sorrendet állítsunk fel közöttük, vagyis kialakítsuk a Bevezetésben már említett, *helyzetfüggő minőségi profilt*. Ebből következni fog, hogy olyan minőségjavítási folyamatot tudunk kidolgozni és bevezetni, amely a minőség minden összetevőjével bizonyos jól meghatározott sorrendben foglalkozik.

A minőségügyi keret alkalmazása akkor lehet sikeres, ha támogatja valamely ismert, elfogadott, netalán kötelezően alkalmazandó megközelítés bevezetését – hiszen ebben az esetben a keret alkalmazója általánosan ismert és elismert elemek, esetleg tanúsítható szabványok alapján szervezi minőségirányítását.

A szakirodalomban rengeteg módszer, módszertan, modell, szabvány, sikeres próbálkozásról szóló beszámoló esettanulmány található, amelyek mindegyike a szoftvergyártás egy bizonyos, valamelyik előbb felsorolt elemével vagy elemcsoportjával foglalkozik.

A *szoftvertermékre* koncentrálnak a korai Boehm ([4]) és McCall-féle ([5]) minőségügyi modell, illetve az ezeket is felhasználó, újraértelmező ISO 9126 szabvány ([6]). A termék bizonyos – biztonsággal kapcsolatos – jellemzőivel foglalkozik az ISO 15408 –as szabvány (más néven „Common Criteria”, lásd [7], [8], [9]).

Folyamatalapú az üzleti okok miatt is igen népszerű ISO 9001:2000 szabvány ([10]), valamint ennek szoftverfejlesztésre való alkalmazását támogató ISO 90003:2004 ([11]), mely kiemelten hangsúlyozza a szoftvertermékhez kapcsolódó mérések fontosságát is.

Elsősorban a szoftvert előállító műszaki, irányítási, támogató és szervezeti szintű *folyamatok* elemeivel foglalkozik a mostanra már második kiadását megélt CMMI modell (lásd ([12], ([13]), mely ötvözi a mára már visszavont, de sokáig igen népszerű CMM modellt (lásd [14]), és a folyamatok érettségét figyelő SPICE modellt / ISO 15504 ([15]) szabványt. A SPICE-ban felhasznált folyamatmodellt az ISO 12207 szabvány ([16]) írja le részletesen.

Magyarországon is egyre több, autóiparhoz kapcsolódó szoftvert gyártó cég alkalmazza a SPICE autóiparra vonatkozó követelményeit, az Automotive Spice-ot ([17]).

Folyamatalapú az EFQM² által ajánlott üzleti kiválósági modell is, mely a teljeskörű minőségmenedzsment elveit helyezi előtérbe.

Az emberi erőforrásokra koncentrálnak a P-CMM ([18]). A szoftverfejlesztés legkritikusabb erőforrásának, az egyéni szoftverfejlesztőknek, valamint a szoftverfejlesztő csapatoknak a teljesítményét hivatott növelni a PSP és TSP – melyek alapvetően folyamatalapú megközelítések (lásd [19], [20], [21]).

Projektmenedzsmenttel (a szoftverfejlesztési projektek irányítási vonatkozásaival) az erre szakosodott módszertanok foglalkoznak kiemelten (pl. PRINCE, lásd [22]).

Az eddig említett szabványok, modellek, megközelítések különböző szinten kezelik a QMIM minőségügyi keret objektumait. Kevés megközelítés adja meg az elemek pontos definícióját. Némelyik meghatároz pontos minőségi attribútumokat bizonyos elemekre,

² EFQM: European Foundation for Quality Management

leginkább a termékre vonatkozóan (pl. a Boehm és McCall modell, valamint az ISO 9126 szabvány). *Mérőszámokra* vonatkozóan az ISO 9126 szabványban találunk termékkel kapcsolatban használható elemeket, de sok más megközelítés is hangsúlyozza azok fontosságát. A Basili-féle GQM paradigma ([23]) a minőségügyi keret bármely objektumához kapcsolódó minőségi attribútumok és a hozzájuk rendelhető mérőszámok meghatározási módjához ad segítséget.

A QMIM minőségügyi keret és az ismert megközelítésének kapcsolata tovább részletezhető (lásd ([2], [3])).

Jelen fejezet további részeiben a fentebb felsorolt megközelítések, szabványok, modellek közül részletesen is bemutatunk néhányat. A különböző modellek bemutatásánál nem törekedtünk teljességre: azokra koncentráltunk, amelyek, tapasztalatunk szerint, jelenleg is alkalmazásban vannak és érdeklődésre tartanak számot a magyarországi szoftvergyártók körében.

6.8.3. A szoftvertermék minőségét hangsúlyozó megközelítések

A szoftver minőségének meghatározásában igen kézenfekvő lehetőség, hogy a vevőnek átadott szoftvertermékben levő hibák számával / súlyosságával próbáljuk leírni a „jó” minőséget. A cél mindig az, hogy felhasználónak átadott rendszer kielégítse az igényeket, és minél kevesebb hibával tegye ezt. Logikus volt tehát az a szoftverfejlesztés „hajnalán” tapasztalt megközelítés, hogy a szoftver minőségét a szoftvertermék minőségével azonosították, és igyekeztek egyértelmű minőségi attribútumokkal leírni azt.

A '70-es években a szoftver minőségét meghatározó elemeket és a közöttük levő kapcsolatokat két szoftverminőség-modell foglalta össze legelőször: a Boehm- illetve a McCall-modell. A Boehm modellt 1977-ben dolgozta ki Barry Boehm, a TRW Systems and Energy Inc. (Thompson Ramo Wooldridge) szoftverházának kutatócsoportjában. A McCall modell (nevét James McCall-ról, a General Electric projektvezetőjétől kapta) 1978-ban került kidolgozásra. A két modell igen hasonló szerkezetet mutat. Legmagasabb szinten felhasználói alapszemponthoz határoztak meg, majd mindegyikhez minőségfaktorokat rendeltek. A minőségfaktorokat szoftverminőség-jellemzőkre bontották, a szoftverminőség-jellemzőkhöz pedig mérőszámokat rendeltek. Mindkét modellben definiálták az alapelemeket, valamint a közöttük való kapcsolatokat, de a két modell elemei csak részben feleltethetők meg egymásnak (a felhasználói alapszemponthoz különbözőek, 6 minőségfaktor és 6 szoftverminőség-jellemző mindkét modellben szerepel, de jócskán vannak a csak egyik vagy csak másik modellben szereplő elemek is). A két modell világszinten ismertté vált, de nem alakult ki konszenzus egyik vagy másik alkalmazását illetően. Bonyolult volt a modellek szerkezete (nagy számú minőségjellemzőt és mérőszámot tartalmaztak, s a mérőszámok, jellemzők és minőségfaktorok közötti kapcsolatok igen szövevényesek voltak).

A szoftveres szakemberek sokáig hiányoltak egy olyan szabványt, amely egyedi, egyértelmű módon határozza meg a szoftvertermék minőségi jellemzőit. Végül, erre az igényre válaszolva, 1991-ben készült el az ISO 9126: „Software Product Evaluation: Quality Characteristics and Guidelines for their Use” szabvány (lásd ([6])), amely felhasználta a McCall és Boehm modelleket. A szabvány jelen formájáig több kiadást is megért, melyekben az eredeti szöveget a felhasználók visszajelzései alapján pontosították.

Az ISO 9126 szabvány célját, előnyeit maga a szabvány is megfogalmazza. Ezek szerint, az ISO 9126 egy jól meghatározott és széles körben elterjedt szoftverminőségi modell iránti igény kielégítésére készült el.

Az ISO 9126 szabvány tulajdonképpen a McCall modell leegyszerűsített változata, amely a szoftvertermék minőséget hat magas szintű jellemzővel, minőségi attribútummal azonosítja: *funkcionalitás, megbízhatóság, használhatóság, hatékonyság, karbantarthatóság és hordozhatóság*. Ez a hat attribútum kerül részletezésre összesen 27 alábontott jellemző bevezetésével. Az alábontott jellemzők és magas szintű jellemzők közötti kapcsolatok világosak és egyértelműek, minden alábontott jellemző egyetlen magas szintű jellemzőhöz kapcsolódik.

Az ISO 9126 szabványt lépésről lépésre finomították. Alkalmazása során kiderült például, hogy a hat jellemzőn kívül olyan jellemzőt is figyelembe kell venni, amely a szoftvertermék valós környezetben történő működését írja le. Ezt a jellemzőt „*működés közbeni minőségnek*” (Quality in use) nevezték el, és 4 alábontott jellemzőt társítottak hozzá.

A 2000-es évek elején került kiadásra a szabványt alkotó összes rész újabb változata. Jelenleg (2008 március) az ISO 9126 szabvány³ a következő részekből áll:

- 1. Rész: Minőségi modell (ISO/IEC 9126-1:2001)
- 2. Rész: Külső metrikák (ISO/IEC TR 9126-2:2003)
- 3. Rész: Belső metrikák (ISO/IEC TR 9126-3:2003)
- 4. Rész: Használat közbeni metrikák (ISO/IEC TR 9126-4:2004).

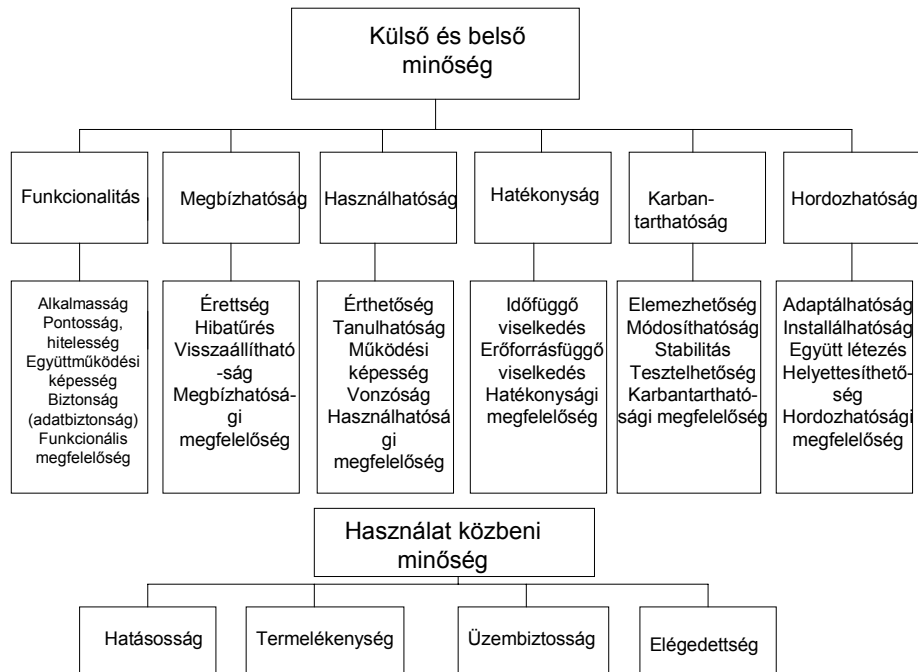
Az 1. rész a szoftvertermékek esetében alkalmazandó minőségi modellt írja le. A modell két részből áll: a belső és külső minőség, illetve a használt közbeni minőség részből.

A modell filozófiája szerint, a szoftverfejlesztési folyamat befolyásolja a termék minőségét, mely (a fejlesztés során) belső, illetve külső minőségi jellemzőkkel írható le. Belső minőségi jellemzők az olyan jellemzők, amelyek magát a szoftver terméket jellemzik, pl. méret. A külső jellemzők függnnek attól a környezettől, amelyben a szoftver működik, vagyis nem mérhetők közvetlenül a szoftveren, csak a működésben lévő szoftveren, pl. hibák száma. Az átadott szoftvertermék bizonyos (felhasználói) környezetben működik, és hatást gyakorol a környezetére. A működő szoftvertermék minősége a működés közbeni minőségi attribútumokkal jellemezhető.

Az ISO 9126 szabványcsalád részletesen is ismerteti a szoftver minőségi jellemzőit és alábontott jellemzőit, a szoftvernek az ún. belső és külső jellemzőit és metrikáit.

Az ISO 9126 szabvány elemeit a 6.8.2. ábra szemlélteti összefoglalóan.

³ Helyesebb lenne „ISO 9126 szabványcsalád”-ról beszélni, a szakzsargon viszont az „ISO 9126 szabvány” elnevezést használja, ezért mi is ezt a nevet említjük.



6.8.2. ábra: Az ISO 9126 minőségi attribútumai, összefoglalás. Forrás: ISO 9126-1

6.8.4. A szoftvergyártás folyamatának minőségét támogató megközelítések

A 6.8.3. fejezetben leírtuk, hogy a szoftverminőség első megközelítései a szoftvertermék minőségi jellemzőire koncentráltak. A '80-as években viszont egyre sürgetőbbé vált, hogy a szoftver minőségét valamilyen „üzleti” szempontú megközelítéssel írják le, amely lehetővé teszi a szoftvert gyártók számára, hogy tevékenységük jó minőségét vevőik felé igazolni tudják. A szoftvergyártó cégeknek kapóra jött az 1987-ben kiadott ISO 9000 szabványcsalád, mely éppen a kereskedelmi tranzakciókban résztvevők számára kínál „üzleti szemléletű” minőségügyi modellt. A szoftverfejlesztők tehát átvették és alkalmazni kezdték az ISO 9000 szabványcsaládot.

Ez a tendencia igen erős volt a '80-as évek végén és a '90-es évek elején – olyan erős, hogy már-már minden egyéb megközelítést kiszorított a szoftverminőség lehetséges megközelítései közül. Ha a szoftvergyártás 6.8.1-es ábrán is feltüntetett fontos objektumaira értelmezzük a váltást, akkor arra a következtetésre jutunk, hogy az ISO 9000 szabványcsalád megjelenésével a szoftverminőség termék alapú megközelítését a folyamat alapú megközelítés váltotta fel. Ebben az időszakban ahelyett, hogy a termékre koncentráltak volna (melynek minőségét meghatározni és objektíven mérni nehéznek bizonyult), az érdekeltek azt az elvet vallották, hogy a szoftverfejlesztés folyamatait kell kézben tartani, s ez garanciája lesz a folyamatok eredményeképpen létrejövő termék jó minőségének. Józan ésszel valóban elvárható, hogy jó folyamat jó terméket hozzon létre, ám a megfelelés nem ennyire egyértelmű.

A '90-es évek közepére azután kiderült, hogy az ISO 9000 szabványcsalád (kiemelten az ISO 9001) alkalmazása csak egy igen kis lépés a szoftverminőség területén. A szoftver

megfelelő minőségének szavatolásához mindenképpen szükség van további megközelítésekre.

Ebben az időszakban alakultak ki a kifejezetten szoftverfejlesztő cégek számára alkalmazható képességi és érettségi modellek. (Másik tendenciaként - mint azt a 6.8.3 fejezetben leírtuk-, az 1990-es években alakították ki a „modern” termék-alapú megközelítést, az ISO 9126-ot, ez azonban, a 2000-es évek első felében történt újrakiadásáig a folyamat-alapú megközelítéseknél sokkal kevésbé vált népszerűvé.)

Ebben a részben a szoftver minőségének folyamat-alapú megközelítését támogató modellek, szabványok közül ismertetjük a leggyakrabban alkalmazottakat.

6.8.4.1. ISO 9001:2000 és ISO 90003:2004

Az ISO 9001:2000 szabvány (mely jelen kiadvány 4. fejezetében részletes bemutatásra kerül) a szoftverfejlesztésre is alkalmazható, ez azonban nem egyszerű feladat. Általában elmondható, hogy az ISO 9001:2000 szabványt csak a szoftverfejlesztés folyamatát, sajátosságait, életciklus modelljeit, eszközeit, technikáit jól ismerő szakemberek tudják *hatékonyan* alkalmazni a szoftverfejlesztésre. Jó megközelítés lehet a szoftverfejlesztő cég folyamatainak azonosításával kezdeni az ISO 9001:2000 –nek megfelelő minőségügyi rendszert. A folyamatok azonosításakor ismert életciklus-modellek (vízesés modell, spirál modell, iteratív fejlesztést támogató modell stb.), szoftverfejlesztési folyamatokat leíró szabványok (pl. ISO 12207), módszertanok (pl. RUP) ajánlásait lehet figyelembe venni. Értékteremtő folyamatok között például azonosítani lehet a szoftverfejlesztésben alkalmazandó sajátos folyamatokat: követelményspecifikáció, elemzés, nagyvonalú tervezés, részletes tervezés, kódolás, dokumentálás, modulteszt, termék integráció, integrációs teszt, rendszerteszt, átadás-átvételi teszt stb. Bár az ISO 9001:2000 nem teszi kötelezővé, a tapasztalat azt mutatja, hogy ajánlatos a szoftverfejlesztés irányítási és műszaki folyamatait írásban is szabályozni.

A szabványalkotók korán felismerték a szoftverfejlesztés sajátosságait, és már 1991-ben (tehát 4 évvel az ISO 9000 szabványcsalád első kiadása után) elkészült az ISO 9001 szoftverfejlesztésre való alkalmazását támogató ISO 9000-3 szabvány (Quality management and quality assurance standard- Guidelines for the application of ISO 9001 to the development, supply and maintenance of software). Az ISO 9000-3 nem volt tanúsítható, tehát csak arra szolgált, hogy az ISO 9001 követelményeinek szoftverfejlesztésre való alkalmazását támogassa.

A '90-es években az ISO 9000-3 szabványt a szoftverfejlesztők széleskörűen alkalmazták a világban. Az Egyesül Királyságban kidolgozták a TickIT módszertant, amely a szoftverfejlesztő cégek ISO 9001 szabvány szerinti tanúsítását tette lehetővé, tartalmazva az ISO 9001 szoftverfejlesztésre való értelmezését és a megfeleltetéseket az ISO 9001 és ISO 9000-3 között. A TickIT módszertant 2001-ben frissítették, és az ISO 9001:2000 szabvánnyal kompatibilissé tették. Jelenleg is alkalmazzák, több új verziója is megjelent (lásd pl. <http://www.tickit.org>).

2000-ben, az ISO 9001:2000 szabvány megjelenésekor paradox helyzet állt elő e szabvány szoftverfejlesztésre vonatkozó értelmezése tekintetében. Visszavonták ugyanis az ISO 9001, ISO 9002 és ISO 9003 szabványokat, de nem vonták vissza az ISO 9000-3-at (amely struktúrájában a visszavont ISO 9001-e követte). A szoftverfejlesztőknek 4

évig kellett várniuk, míg megszületett az új, struktúrájában már az ISO 9001:2000 szabványt követő, annak szoftverfejlesztésre való értelmezését támogató ISO / IEC 90003: 2004 szabvány (lásd [11]). Jelenleg tehát ez a – nem tanúsítható - szabvány ad útmutatást arra nézve, hogyan alkalmazzuk az ISO 9001:2000 szabványt szoftverfejlesztésre. A termék és folyamat minőségét hangsúlyozva, ez a szabvány felhasznál ISO 9126 és ISO 12207 (lásd 6.8.3 és 6.8.4.2) elemeket egyes fogalmak magyarázatára, követelmények pontosítására (pl. azt ajánlja, hogy a szoftvertermék minőségi attribútumainak meghatározására és mérésére alkalmazzuk az ISO 9126 szabványt). Az ISO 90003: 2004 szabvány (2005 júniusától magyar nyelven is rendelkezésre áll) meghatározásai szoftver-specifikusak. (Pl. a regressziós tesztelés fogalmát a következőképpen határozza meg: „az a tesztelés, amely azt hivatott megvizsgálni, hogy a rendszer egy komponensének módosulása nem hatott-e negatívan a funkcionalitásra, megbízhatóságra vagy a teljesítményre, vagy nem vitt-e be további hibát a rendszerbe.”) Az ISO 9001:2000 struktúráját az ISO 90003:2004 több helyen további elemekkel (alfejezetek, magyarázatok) egészíti ki.

6.8.4.2. Az ISO 12207 szabvány

A szoftverfejlesztési folyamatokat leíró szabványok között fontos az eredetileg 1995-ben megjelent (majd, a 2000-es évek elején kiegészített) ISO/IEC 12207:1995 szabvány (magyarul 2000-től áll rendelkezésre, lásd [16]).

Ez a szabvány egységes fogalmi keretet hoz létre a szoftveréletciklus-folyamatokra olyan jól meghatározott terminológiát használva, amely a szoftveripar számára kiindulásul szolgálhat. Olyan folyamatokat, tevékenységeket és feladatokat tartalmaz, amelyek szoftvert tartalmazó rendszerek, különálló szoftvertermékek és szoftverszolgáltatások beszerzése, valamint szoftvertermékek szállítása, fejlesztése, üzemeltetése és karbantartása során alkalmazandók. Alkalmazható rendszerek, valamint szoftvertermékek és szoftverszolgáltatások beszerzésénél, szoftvertermékek és förmver⁴ szoftverrészének szállítása, fejlesztése, üzemeltetése és karbantartása során, függetlenül attól, hogy azt a szervezetten belül vagy kívül hajtják végre.

Az ISO 12207 szabványt két felet érintő helyzetekben történő használatra tervezték, de alkalmazható akkor is, ha a két fél ugyanabból a szervezetből való. A szabvány olyan folyamatokat, tevékenységeket és feladatokat tartalmaz, amelyeket úgy alakítottak ki, hogy a szoftverprojekt esetén illeszthetők legyenek. Az illesztés a nem alkalmazandó folyamatok, tevékenységek és feladatok törlését jelenti.

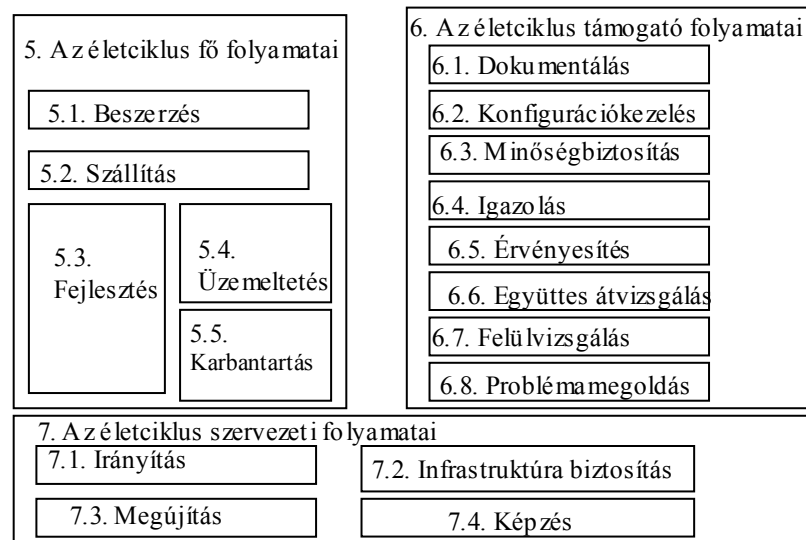
E szabványnak való *megfelelés* azt jelenti, hogy végrehajtották az összes olyan folyamatot, tevékenységet és feladatot, amit a szoftverprojekt céljaira az illesztési folyamat segítségével ebből a szabványból kiválasztottak.

Minden olyan szervezetnek, amely e szabványt kereskedelmi feltételként előírja, kötelessége, hogy meghatározza és nyilvánosságra hozza a folyamatok, tevékenységek és feladatok azon legszűkebb körét, amely megfogalmazza a szállítók e nemzetközi szabványnak való megfelelését.

⁴ Förmver: Hardverberendezés és számítógépes utasítások vagy adatok olyan kombinációja, amely csak olvasható szoftverként jelenik meg a hardverberendezésben. A szoftvert program segítségével nem lehet könnyen módosítani. (ISO12207)

Az ISO 12207 szabvány korlátja, hogy leírja ugyan a szoftverélekciklus-folyamatok szerkezetét, de nem adja meg a folyamatokban szereplő tevékenységek és feladatok megvalósításának vagy végrehajtásának részleteit, nem írja elő a dokumentáció nevét, formáját, konkrét tartalmát, nem ír elő semmilyen konkrét élelciklus modellt vagy szoftverfejlesztési módszert.

Az ISO 12207 szabvány 5. , 6. és 7. fejezetei mutatják be a szoftver-élelciklus folyamatokat. E fejezetek szerkezetét a 6.8.4. ábra szemlélteti.



6.8.4.. ábra: Az ISO 12207 szabvány szerkezete

Az ISO 12207 szabvány átmenetet képez az üzleti szemléletű minőségirányítási szabványok és a technikai jellegű, szoftverminőséghez kapcsolódó szabványok között. Érdeemes megemlíteni, hogy az ISO 12207 szabvány minden fejezetének megfeleltethető a szoftverfejlesztéshez kapcsolódó technikai részleteket leíró „V-modell”.

A V-modellt 1997-ben az informatikai rendszerek fejlesztésének élelciklusát leíró szabvány keretében tették közzé (lásd [24]), az akkori Német Szövetségi Köztársaságban, a Szövetségi közigazgatás számára. Ezek szerint a V-modell „a rendszerfejlesztés, módosítás és karbantartás folyamán szereplő összes tevékenységet, terméket, valamint kapcsolatukat szabályozza”. Elsősorban az IT-vel kapcsolatos feladatokra koncentrál.

Az ISO 12207 szolgált alapul a következő részben bemutatott képesség-érettségi modellek folyamat-dimenziójának kialakításához.

6.8.4.3. Érettségi és képességi modellek

A szoftverfejlesztő szakemberek nem elégedtek meg az ISO 9001 által felkínált minőségi megközelítéssel. A továbblépésre az adta az impulzust, hogy nagy (és elsősorban az amerikai hadiiparhoz kapcsolódó) cégek szerették volna szoftverbeszállítói munkáját átláthatóvá tenni, csökkentve ezáltal annak kockázatát, hogy a beszállítandó termék nem készül el időben, vagy nem lesz megfelelő minőségű. Az 1980-as évek végén kutatás indult annak vizsgálatára, hogy milyen módszerrel lehetne „osztályozni” a szoftvergyártó cégeket.

Hamar kiderült, hogy egyes szervezetek „érettebbek” mint mások, s ez elsősorban a „fegyelmezett” folyamat-végrehajtásban érhető tetten. Az érett szervezetek nagyobb valószínűséggel képesek időre leszállítani a megfelelő minőségű szoftvert.

Felmerült tehát a kérdés: milyen jellemzők mutatják egy szervezet érettségét? Kiderült, hogy egy szervezetben annál valószínűbb a sikeres szoftverfejlesztés, minél több, jól szervezett folyamatot működtet a szervezet. A szoftver folyamat érettsége, Paulk szerint (lásd [14]), annak mértéke, hogy egy folyamat mennyire pontosan meghatározott, vezérelt, mért, ellenőrzött és hatékony. Az érett szoftver folyamat tehát *meghatározott (definiált), vezérelt, mért, ellenőrzött, hatékony és javulásra képes*. A folyamat annál érettebb, minél több elem van a helyén az előbb felsoroltak közül. A szervezet annál „érettebb”, minél több „érett” folyamatot alkalmaznak a szervezetben.

Az évek során az a konvenció alakult ki, hogy *szervezet érettségéről és folyamatok képességéről* beszélünk. Megjegyzendő azonban, hogy a két elnevezés gyakran keveredik a mindennapi szóhasználatban. Beszélünk például „képesség-érettségi modellekről” – ami teljesen logikus elnevezés, hiszen a szervezet érettségét a folyamatok száma és azok képességi szintje mutatja.

Felismerve, hogy a képességi-érettségi modelleket a szoftvergyártásban részt vevő folyamatok fejlesztésére használják (ebben, áttételesen, a szervezeti érettség fejlesztése is benne van), a szakirodalomban szokássá vált a képességi-érettségi modelleket *folyamatfejlesztési* vagy *folyamatjavítási* modellként is emlegetni. Attól függően, hogy a fejlődés szakaszosan (lépcsőkben), vagy folyamatos, kisebb lépésekben történik, a folyamatfejlesztési megközelítések, modellek csoportokba sorolhatók.

A folyamatjavítási modellek között vannak lépcsős modellek (staged models), pl. CMM, folytonos modellek (continuous models), pl. a SPICE, Automotive SPICE, és integrált modellek, pl. CMMI[®].

A lépcsős modellek a teljes szervezetet vizsgálják. Foglalkoznak: vezetési és műszaki folyamatokkal, az alkalmazott technológiával, magával a szervezettel. Adott érettségi szinten a szervezetnek bizonyos, a modellben előre definiált) folyamatokat kell működtetnie.

A folytonos modellek az egyes folyamatokra (és nem a teljes szervezetre) koncentrálnak, folyamatokra állapítanak meg képességi szinteket bizonyos jellemzők alapján. Az ilyen modell alkalmazója maga döntheti el, hogy milyen folyamat érettségét szeretné vizsgálni. A „kombinált”, integrált modellek ötvözik a kétféle modellt, a bizonyítottan hasznos elemeket kiválasztva. A szervezet bizonyos érettségéhez bizonyos folyamatoknak bizonyos, előre meghatározott képességi szintje szükséges.

A továbbiakban mindegyik kategóriából egy-egy igen ismert modellt mutatunk be: a lépcsős CMM modellt, a folytonos SPICE modellt és az integrált CMMI[®] modellt. Megjegyezzük, hogy a CMM modell 2005 decemberétől hivatalosan már nem auditálható, és fokozatosan visszavonásra került, helyét a CMMI[®] modell vette át. Mivel

azonban a modell igen népszerű volt az 1990-es években, és alapul szolgált a CMMI[®] modell kifejlesztésekor, fontosnak tartjuk bemutatni. Jelenleg (2008 március) a CMM modellt még több nagy cég alkalmazza, belső folyamatfejlesztésre.

6.8.4.3.1. A CMM modell

A CMM (Capability Maturity Model) egy, szoftverfejlesztő cégek számára kidolgozott, lépcsős folyamatfejlesztési modell.

A beszállítói problémára megoldást keresve, 1984 decemberében a DoD (Department of Defence) támogatásával létrejött a Software Engineering Institute (SEI⁵) a Carnegie Mellon Egyetemen. Ez az intézet azóta világelsőséget és – monopóliumot vívott ki magának a szoftverfolyamat-fejlesztési modellekkel kapcsolatos szolgáltatásokban.

A SEI-ben 1986-tól projekt indult szoftverfolyamat-fejlesztési témában, melynek keretében 1989 és 1991 között, Watts Humphrey vezetésével, kidolgozták a CMM modellt.

A CMM modell érettségi szintjei a következők:

- 1. kaotikus vagy kezdeti szint (Initial)
- 2. ismételhető szint (Repeatable)
- 3. meghatározott szint (Defined)
- 4. menedzselt szint (Managed)
- 5. optimalizáló szint (Optimizing).

Az érettségi szintek jellemzői, röviden, a következők:

- 1., kaotikus v. kezdeti szint: nincsenek folyamatok, mindenki tűzoltásszerűen dolgozik, nagy a kockázat és szinte nem létező az ellenőrzés a projekteknél, ha elkészülnek, az egyének zsenialitása, hősiessége miatt készülnek el.
- 2., ismételhető szint: az egyes emberek a saját munkájukat ismételni tudják. A projektmenedzsment megfelelő, projektirányítási rendszer működik a cégnél. A projekteket tervezik, követik és vezérlik, de az eljárások, szokások projektenként különbözhetnek. Az ezen a szinten működő folyamatok a következők: projekttervezés, projektkövetés- és vezérlés, követelménymenedzsment, minőségbiztosítás, beszállítók kezelése, konfigurációmenedzsment.
- 3., meghatározott szint: a folyamatok cég szintjén szabályozottak, az egyes projektek saját folyamataikat a szabványos folyamatokból, testreszabási útmutatókat használva, szabják teste. A projektirányítási eljárásokon kívül megvannak a dokumentált műszaki eljárások is. Meghatározták és dokumentálták a cégnél alkalmazott életciklus modelleket. Becsülik, mérik, kézbentartják a szoftvertermék méretét, valamint a termék- és folyamat-minőség elfogadható értékeit. Ezen a szinten – a 2-es érettségi szint folyamatain kívül - a következő folyamatok vannak jelen: szervezeti szintű folyamatszeglés, szervezeti szintű folyamat-meghatározás, szoftvertermék-fejlesztés, integrált szoftvermenedzsment, páros (egyenrangú) szemlék, képzési program.
- 4., menedzselt szint: a folyamatokat mérik, a mérési eredményeket statisztikai módszerekkel elemzik. Ezen a szinten – a 2-es és 3-as érettségi szint folyamatain

⁵ A SEI web-lapja: www.sei.cmu.edu

kívül - a következő folyamatok vannak jelen: mennyiségi projektmenedzsment, szoftverminőség-menedzsment.

- 5., optimalizáló szint: a mérések eredményeit folyamatos optimalizálásra használják fel. A technológiai változásokat tervezik és folyamatosan követik. Ezen a szinten – a 2-es, 3-as és 4-es érettségi szint folyamatain kívül - a következő folyamatok vannak jelen: folyamat-változásmenedzsment, technológiai változásmenedzsment, hibamegelőzés.

A CMM modellt széles körben alkalmazták a világ minden részén. A SEI által közzétett adatokból megtudjuk, hogy 1987 és 2005 között a világban összesen 2794 felmérést végeztek, összesen 1367 szoftvercég 16290 projektjét mérték fel. A szervezetek érettségi szintjének átlaga emelkedett az évek során. 2005-ben az összes korábbi felmérés alapján, a felmért szervezetek 5,7%-a volt 1-es érettségi szinten, 39,6%-uk 2-es érettségi szintű, 37,4%-uk 3-as érettségi szintű, 7,6%-uk 4-es érettségi szintű, 9,8%-uk pedig 5-ös érettségi szintű besorolást kapott.

A CMM modellt sok kritika is érte az évek folyamán. Legerősebben azt kifogásolták a modell bírálói, hogy a mérések végzése csak a 4-es érettségi szinten jelenik meg követelményként, holott, szerintük, bármely folyamat csakis mérésekkel tartható kézben. A kritikára sokan és sokszor válaszolták, hogy méréseket már alacsonyabb szinten, folyamatosan végezni kell, ezt a CMM „nem tiltja meg”. Mindenesre, a CMM modellt felváltó CMMI[®] modellben a mérési és elemzési folyamatot a lépcsős megközelítés 2-es szintjén teszik kötelezővé.

6.8.4.3.2. A SPICE modell /ISO 15504 szabvány

A SPICE a „Software Process Improvement and Capability dEtermination” rövidítése, és azt a nemzetközi projektet jelöli, amelynek keretében 1993-1998 között kidolgozták a később ISO 15504 szabványcsaládban közzétett átfogó referencia- modellt a szoftverfejlesztési folyamatokra és a folyamatok képességi szintjére vonatkozóan, kis-, közepes- és nagyvállalatok nemzetközi tapasztalatait összegezve. (Bár helyesen „SPICE projektet” és „ISO 15504” szabványt kellene emlegetnünk, a szakzsargonban gyakran „SPICE modell” névvel illetjük azt a keretrendszert, amely a SPICE projekt keretében, az ISO 15504 szabványcsaládban került kidolgozásra. A „szokásnak” megfelelően, a továbbiakban én is alkalmazom a „SPICE modell” elnevezést. A szerző.)

A SPICE modell folytonos megközelítésű. A cég egyes szoftverfejlesztési folyamatait vizsgálja, melyek képessége 0 és 5 szint között mozoghat. A folyamat képessége annál magasabb, minél több jellemző van a helyén a tervezés, dokumentálás, mérés, folyamatos fejlesztés attribútum-csoportokból. A SPICE modellt olyan cégeknek ajánlatos alkalmazni, amelyek bizonyos, 1-től különböző érettségi szinten vannak a cég egészére vonatkozóan, és bizonyos folyamataikat szeretnék továbbfejleszteni.

Az ISO 15504 első változatát 1998-ban adták ki, majd, a kiadását követő időszakban a SPICE projekt-vezetőség (mely öt nemzetközi Műszaki Központ szövetségéből jött létre) irányította a szabvány kipróbálását, tesztelését, és a visszajelzések alapján történő pontosítását. E tevékenység eredményeképpen a SPICE modellt (a kapcsolódó szabványban leírt keretrendszert) 20 ország szoftverfejlesztői, szabványügyi szakértői és kutatói tesztelték és véleményezték. Kezdetét vette a SPICE auditorok képzése is, a

SPICE –keretrendszer szerint végzett felmérések pedig növekvő népszerűsége tettek szert.

A népszerűség oka valószínűleg abban keresendő, hogy a SPICE igen jól alkalmazható szoftverfolyamat-fejlesztésre. Ezzel az adott időszakban komoly versenytársra vált a többi, piacon levő érettségi modell között. A korábban kifejlesztett CMM-re ugyanis rányomta bélyegét az a tény, hogy az Amerikai Honvédelmi Minisztérium beszállítóinak értékelésére hozta létre (bár kétségtelen, hogy a CMM első változatát követő frissítések igyekeztek eltávolodni ettől a szemlélettől és támogatni a folyamatfejlesztést). Az egyéb érettségi modellek a British Telecom, Bell Canada / BNR és Bellcore, termékei lévén, főleg a telekommunikációban váltak elterjedté. Fontos jellemzője még a SPICE-nak, hogy nemzetközi szabványként kerül leírásra, emiatt a világon bárhol ismert, és az alkalmazó cég „hovatarozásától” függetlenül használható. Ezzel együtt elmondható, hogy a SPICE modellt inkább német nyelvterületen alkalmazzák, és elsősorban az autópárházban terjedő, Automotive Spice „al-változata” igen népszerű (erre vonatkozóan lásd a 6.8.4.3.2.1 Fejezetet.)

Az „ISO/IEC TR 15504 Software Process Assessment” dokumentumok összességét általában SPICE99-ként emlegetik. Ennek kiadását követően a szabványtervezet gyökeres változáson esett át. Egyrészt címében a folyamatjavítás mellől elmaradt a „szoftver” szó, ezzel tetszőleges folyamatok felmérésére alkalmassá vált, másrészt részei is alapvetően átalakultak. A dokumentumcsomag jelenleg (2008 márciusában) a következőket tartalmazza (csak angolul állnak rendelkezésre, lásd [15]):

- ISO/IEC 15504-1:2004: Part 1 : Concepts and vocabulary
- ISO /IEC 15504-2: 2003: Part 2 : Performing an assessment
- ISO /IEC 15504-3: 2004: Part 3 : Guidance on performing an assessment
- ISO /IEC 15504-4:2004: Part 4 : Guidance on use for process improvement and capability determination
- ISO /IEC 15504-5: 2006: Part 5 : An exemplar process assessment model for software life cycle processes
- ISO /IEC 15504-6: 2007: Part 6 : An exemplar process assessment model for system life cycle processes
- ISO /IEC 15504-Part 7 : Assessment of organizational maturity – megjelenése 2008-ban várható
- ISO /IEC 15504-Part 8:2007 : Exemplar IT Service Management Process Assessment Model (NWI Ballot October 2007)

A fensorolt címekből is látható, hogy az ISO / IEC 15504 szabvány első öt része általános keretrendszert fogalmaz meg szoftverfolyamatok értékelésére vonatkozóan. Eredetileg nem terveztek konkrét folyamat-modellt leírni, hanem azokat az elemeket hangsúlyozták, amelyek egy szoftverfejlesztő cég folyamatainak azonosításában, jellemzőik megértésében és a folyamatok képességi szintkének értékelésében használhatók. Sokáig a SPICE model csak ajánlást tartalmazott arra nézve, hogy a szoftverfejlesztési folyamatokat az ISO 12207 szabványnak megfelelően lehetséges azonosítani. A modell ebben az állapotában is ismertté vált, és sokan alkalmazták sikerrel szoftvercégek folyamatainak fejlesztésében. A megközelítést szimpatikussá tette, hogy nem határozott meg a cégek számára kötelezően alkalmazandó folyamatokat, hanem a felhasználókra bízta a fejlesztési kívánt folyamatok kiválasztását. Ez a lehetőség azonban nem mindenki számára előny: alacsony érettségi szinten levő szoftvercég ugyanis nem biztos, hogy

képes felmérni, mely folyamatokkal szükséges foglalkoznia a káoszról kivezető út elején. Ebből a szempontból a CMM talán hasznosabb: alacsony érettségi szinten levő cégek számára egészen pontosan megfogalmazza, mely folyamatokkal érdemes foglalkozni, hogy a cégnél kialakuljon a későbbi folyamatfejlesztést lehetővé tevő "rend".

A SPICE modellel foglalkozó szakértők 2006-ban mégis kiadták a szabvány 5. részét, amely egy példa-értékű szoftverfejlesztési folyamatmodellt mutat be. Az ISO 15504 szabványnak ez a része szintén a korábban ajánlott, ISO 12207 szabvány folyamataira támaszkodik.

Az ISO / IEC 15504 szabvány 6, 7, 8 részek kidolgozására vonatkozóan csak a szabványcsalád első öt részének kiadását követően született meg a döntés. Az eddig megjelent, 6. és 8. részek a rendszerfejlesztésben és az információtechnológiai szolgáltatásokban alkalmazható folyamatokat ismertetik. A tervezett 7. rész kapcsolatot fog teremteni a folyamatok képességi szintjét vizsgáló folytonos megközelítés és a szervezetek érettségét vizsgáló lépcsős megközelítés között.

A SPICE projekt előrehaladásáról, valamint a nemzetközi szabvány kialakulásáról és kiadásáról további részleteket tartalmaznak Rout (lásd [25], [26]) és Jung & all (lásd [27]) munkái.

A SPICE projekt hivatalosan lezárásra került. Ezzel egy időben létrehozták a SPICE Network-ot, amely a modell / szabvány további felhasználását támogatja. A hálózat célja, például, hogy a SEI-vel hasonló módon támogassa és irányítsa a SPICE-nak megfelelő auditokat, s adjon lehetőséget a hivatalos SPICE felmérések adatait közös adatbázisban tárolni (jelenleg ez még nem megoldott).

A SPICE legfőbb támogatói a világ SPICE-felhasználóit tömörítő *SPICE User Group*, amely évente SPICE-konferenciát is szervez, valamint az iNTACS (International Assessor Certification Scheme), amely a SPICE modell szerint auditálók képzését és akkreditálását tűzte ki célul.

A SPICE-szal kapcsolatban a következő Internet-címen lehet a legtöbb hasznos információt megszerezni: <http://www.isospice.com/>. A továbbiakban összefoglaljuk a legfontosabb tudnivalókat a SPICE szerkezetéről.

A SPICE modellnek két dimenziója van: a folyamat-dimenzió, és a képesség dimenzió.

A *folyamat-dimenzió* a szoftverfejlesztő cégeknél megjelenő folyamatok és jellemzőik meghatározásával foglalkozik. A folyamatok alapvetően az ISO 12207-ben leírt folyamatok. A folyamatokat csoportokba (kategóriákba) sorolják aszerint, hogy felhasználóval kapcsolatosak-e, illetve fejlesztési, támogató, irányítási vagy szervezeti szintű folyamatok-e.

- Az Ügyfél-beszállítói (Customer, CUS) folyamatok azon folyamatok, melyek közvetlen hatással vannak az ügyfélre, support és szoftver szállítása az ügyfélnek, a szoftvertermék és/vagy szolgáltatás helyes működéshez járulnak hozzá.
- A Műszaki, fejlesztési (Engineering, ENG) folyamatok olyan folyamatokból áll, melyek közvetlenül specifikálják, implementálják vagy karbantartják a szoftverterméket, illetve annak a rendszerhez való kapcsolatát és felhasználói dokumentációját. Abban az esetben, amikor a rendszer teljes egészében szoftverekből épül fel, a Műszaki folyamatok a szoftver felépítésével és karbantartásával foglalkoznak.

- A Támogató (Support, SUP) folyamatok azok a folyamatok, melyeket bármilyen más folyamat alkalmazhat (beleértve más support folyamatokat is) a szoftver életciklusának különböző pontjain.
- A Menedzsment (Managerial, MAN) folyamatok azon folyamatok, melyek olyan általános gyakorlatokat tartalmaznak, amelyeket bárki felhasználhat, aki a szoftver életciklusában bármilyen típusú projektet vagy folyamatot menedzsel.
- A Szervezeti (Organisational, ORG) folyamatok segítenek megvalósítani a szervezet üzleti céljait és azon folyamat-, termék- valamint az erőforrás - eszközöket fejlesztik, melyek a projektek során szervezet üzleti céljainak elérésében segítenek.

A *képesség-dimenzió* a kiválasztott folyamatok fejlettségét mutatja. A modell szerint, a folyamatokat a jól meghatározott input és output, a jól megfogalmazott cél, a folyamat tevékenységeinek pontos leírása, a tevékenységek elvégzéséhez szükséges felelőségek, hatáskörök és kompetenciák megléte, a folyamat követésére beiktatott ellenőrzési pontok, a pontosan megfogalmazott teljesítmény-elvárások, a folyamat mértékek jellemzik. Minél több elem „van a helyén” egy adott folyamat esetében, annál magasabb az adott folyamat képességi szintje.

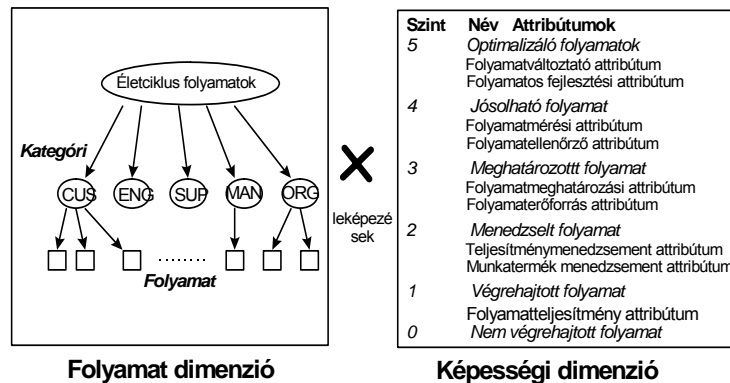
A SPICE-ban azonosított, egyes folyamatokra vonatkozó képességi szintek a következők:

- 0. szint: nem végrehajtott folyamat. Ebben az esetben a folyamatot nem hajtják végre, következésképpen jellemzői sincsenek.
- 1. szint: végrehajtott (folyamat): a folyamat létezik, végrehajtják a szervezetben, de azonosítható jellemzői nincsenek. Nagy a valószínűsége annak, hogy az adott folyamatot a különböző projektekből a különböző emberek esetenként különbözőképpen hajtják végre.
- 2. szint: menedzselte folyamat: a folyamatot végrehajtják, és menedzselésével kapcsolatosan léteznek ismérvek. A folyamatra vonatkozóan van teljesítmény menedzsment, ami azt feltételezi, hogy megtörténik az erőforrás igények meghatározása, a folyamat teljesítményének tervezése, megvalósul a tervezett tevékenységek implementálása és a tevékenységek elvégzésének menedzselése. Megfelelő a munka eredményének menedzselése is. Megtörténik az integritásra és minőségre vonatkozó követelmények meghatározása, a folyamatban szükséges tevékenységek meghatározása, megvalósul a munka eredményének konfigurációkezelése és a munka eredményének minőségmenedzsmentje.
- 3. szint: meghatározott folyamat: a folyamatot a szervezetben meghatározták (dokumentálták), elkészült a folyamat szervezeti szinten érvényes szabványos leírása, és a sajátos esetekben a folyamatot a szabványos folyamatból szabják testre, a szintén dokumentált testreszabási útmutatók alkalmazásával. A meghatározott folyamatot a szervezet egészében bevezették, és gyűjtik a visszajelzéseket a szabványos folyamatra vonatkozóan. Magasabb szinten valósul meg a folyamathoz rendelt erőforrások kezelése is. Megtörténik az emberi erőforrás kompetenciájának meghatározása, a folyamat infrastrukturális követelményeinek meghatározása, megfelelő képességű emberi erőforrások biztosítása és a megfelelő infrastruktúra biztosítása.
- 4. szint: jósolható folyamat: a rendelkezésre álló adatok, tapasztalatok alapján a folyamat végrehajtásának módja és teljesítménye jósolhatóvá válik. Megfelelő a folyamat mérése: megtörténik a folyamatok céljainak és a kapcsolódó mérőszámoknak a meghatározása, a méréshez szükséges megfelelő erőforrások és infrastruktúra biztosítása, megvalósul a meghatározott mérési adatok gyűjtése s ezek

alapján annak figyelése, hogy a folyamat céljai teljesültek-e. Magasabb szinten valósul meg a folyamat ellenőrzése: elemzési és ellenőrzési technikák kerülnek meghatározásra, megfelelő erőforrásokat és infrastruktúrát biztosítanak a mérésre, megfelelő a meglévő mérési eredmények elemzése, az eltérések azonosítása és a szükséges beavatkozás.

- 5. szint: optimalizáló folyamat: a folyamatot állandóan figyelik, és a tényleges teljesítmény alapján folyamatosan fejlesztik. A folyamat változása tervezett és kézbentartott, amennyiben a szabványos folyamatban szükséges változásokat azonosítják és jóváhagyják, a folyamat-változás bevezetéséhez szükséges erőforrásokat rendelkezésre bocsátják, a jóváhagyott változásokat bevezetik, és vizsgálják a változtatás hatékonyságát. A szervezetben cél a folyamat folyamatos javítása, ami a javítási lehetőségek azonosítását, a bevezetési stratégia meghatározását, a testre szabott folyamat meghatározott területén végrehajtott módosítás bevezetését és a változtatás hatékonyságának vizsgálatát jelenti.

A SPICE modellben tehát két dimenzió létezik: a folyamat-dimenzió, és az egyes folyamatokra vonatkozó képesség-dimenzió. A SPICE-nak megfelelő képesség-vizsgálat e két dimenzió elemeinek kereszt-szorzatából áll össze.



6.8.5 ábra: Az SPICE (ISO 15504) referencia modell dimenziói

A „kereszt-szorzat” azt jelenti, hogy bármely folyamatra vizsgálhatunk képességi szintet. A modell alkalmazásához meg kell érteni egyrészt a szoftverfejlesztési folyamat, illetve a szoftverfejlesztő szervezetben meglévő folyamatok elemeit, ki kell választani a felmérni kívánt folyamatokat, majd meg kell vizsgálni az egyes folyamatok képességét.

A SPICE-nak megfelelő képesség-felméréshez szükség van egy felmérési modellre, egy felmérési módszerre és szakképzett auditorokra. Az ISO /IEC 15504 szabvány 2. és 3. része leírja a felmérési modellt, valamint a felmérési módszerrel szemben támasztott követelményeket. Ezek alapján a felmérést végzők dolgozzák ki saját felmérési módszertanukat. Ezek a módszertanok összemérhető eredményeket fognak szolgáltatni (hiszen mindegyikük megfelel a szabvány követelményeinek).

A SPICE-felmérés tulajdonképpen azt jelenti, hogy kiválasztott folyamatok esetében vizsgálják azok képességi szintjét. Minden folyamatnak ki kell elégíteni a sajátosságaiból eredő követelményeket – ezek az ún. „sajátos célok” (specific goals). (pl. a

konfigurációkezelésnek biztosítani kell, hogy a konfiguráció elemeit azonosítsák, minden projekt tag a megfelelő verziót használja, a változásokat megfelelően követik stb.) A sajátos célok teljesülése tulajdonképpen azt jelenti, hogy a folyamatot végrehajtják (tehát legalább 1-es képességi szintű).

A továbbiakban azt vizsgálják, hogy a folyamat esetében teljesülnek-e, és milyen mértékben az általános célok: a folyamatnak bizonyos általános célokat ki kell elégítenie, és bizonyos munka - eredményeket (termékeket) kell létrehoznia. Az általános célok teljesülési foka mutatja a folyamat képességét (ami tulajdonképpen azt jelenti, hogy mennyire „intézményesítették” a folyamatot a szervezetben).

A SPICE-felmérést képzett, regisztrált SPICE-auditorok végzik. Munkájukat kérdőívek segítik. A folyamatok képességére vonatkozó információkat megbeszéléseken, interjúkon nyerik a szervezet tagjaitól. Az elmondottak bizonyítására az auditorok esetenként dokumentumokat, egyéb munkatermékeket is ellenőriznek. A megbeszélésekről emlékeztetők születnek, majd elkészül a felmérés eredményét összefoglaló jelentés. A jelentésnek lényeges eleme a további fejlődési irány ismertetése.

6.8.4.3.2. 1. Sajátságos SPICE-modellek: az Automotive SPICE

A SPICE – tulajdonképpen konkrét folyamatoktól független – képességi és folyamatfejlesztési sémája olyan sikeresnek bizonyult (elsősorban német nyelvterületen), hogy bizonyos ágazatok „saját, ágazat-specifikus SPICE modell” kidolgozásába kezdtek. Az „ágazat-specifikusság” azt jelenti, hogy az adott ágazatban sajátos folyamatokat részesítik előnyben, miközben igyekeznek megtartani az „általánosan érvényes” SPICE folyamatokat is.

•Jelenleg (2008 március) négy kezdeményezésről tudunk, melyek közül 2007 végén kiadásra az autógyártásban alkalmazott Automotive SPICE (www.automotivespice.com) kiadásra is került. Munkacsoportot alakítottak még a banki környezetben alkalmazható Banking SPICE (www.bankingspice.com), valamint a vállalati szintű integrált rendszert alkalmazó környezetet támogató SPICE (www.enterprisespice.com) létrehozására. Ezek munkacsoportok tevékenysége 2008 első félévében fog elkezdődni. A SPICE honlapon szerepel még a MediSPICE munkacsoport is, ennek munkájáról azonban semmit sem sikerült kideríteni.

•A továbbiakban bemutatjuk a Magyarországon is egyre ismertebb és népszerűbb Automotive SPICE modellt.

Az Automotive SPICE modellt az autógyártók egyik érdekképviselője, az Automotive Special Interest Group (SIG) dolgozta ki az Automotive SPICE™ projekt keretében, a korábbi SPICE modellt és az ISO 15504 szabványt kidolgozó SPICE User Group-pal együttműködve.

Az Automotive SIG tagjai között van az AUDI AG, a BMW Group, a DaimlerChrysler AG, a Fiat Auto S.p.A., a Ford Werke GmbH, Jaguar, a Land Rover, a Porsche AG, a Volkswagen AG és a Volvo Car Corporation.

Az Automotive Spice két részből áll: Automotive Spice Process Reference Model (PRM) és Automotive Spice Process Assessment Model (PAM). Csak angol nyelven hozzáférhető.

A Process Reference Model (PRM) modellt 2005 augusztusában adták ki először. Jelenleg a modell v4.3 verziója, 2007.05.05-i kiadása van érvényben. Ez a rész kompatibilis az ISO 15504-2 követelményeivel, és definiálja az Automotive SPICE által alkalmazott folyamatokat: 31, az ISO 12207 szabványban meghatározott folyamatot tartalmaz.

A Process Assessment Model (PAM) modell első kiadása szintén 2005 augusztusi, jelenleg a modell 2007.05.05-én kiadott, v2.3 verziója van érvényben. Ez a rész az ISO/IEC 15504-2-nek mindenben megfelelően határozza meg az autóiipari beszállítók körében alkalmazható folyamat-értékelési modellt.

A SPICE és Automotive SPICE közötti különbség csak az értékelendő folyamatok halmazában van. Az Automotive SPICE nagy hangsúlyt fektet az autóiipari beszállítók szoftverfejlesztési tevékenységének ellenőrzését, értékelését lehetővé tevő folyamatokra. A folyamatokat a következő csoportokba sorolja:

- elsődleges életciklus folyamatok csoportja: beszerzési folyamatok (ACQ, 7 folyamat), értékesítési folyamatok (SUP, 2 folyamat), fejlesztési folyamatok (ENG, 10 folyamat)
- szervezeti folyamatok: menedzsment folyamatok (MAN, 3 folyamat), folyamatfejlesztési folyamatok (PIM, 1 folyamat), újrafelhasználással kapcsolatos folyamatok (REU, 1 folyamat),
- támogató folyamatok csoportja (egyetlen csoport, SUP, 7 folyamat).

A folyamatok modellbeli leírása tartalmazza a folyamatok egyes funkcionális céljait. Mindegyik folyamathoz pontosan meghatározott eredmények tartoznak, melyek megléte igazolja a folyamat sikeres végrehajtását.

Az Automotive SPICE képesség-dimenziója megegyezik a SPICE modell képesség dimenziójával, azzal a különbséggel, hogy a folyamatok képességi szintjét jellemző általános gyakorlatokon kívül általános erőforrások is társulnak az adott képességi szintekhez.

A SPICE modell filozófiájának megfelelően, az Automotive SPICE egy, az ISO 15504 szabvány 2. és 3. részének megfelelő modell és módszertan alapján auditálható.

A német autógyártók egy csoportja (Audi, BMW, DaimlerChrysler és VW) létrehozta a Hersteller Initiative Software (HIS) nevű szervezetet, amely az Automotive SPICE-ban leírt folyamatok közül kiválasztotta azt az eredetileg 11, mostanra 15 folyamatot, melyet a szervezet minimálisan felméréndő folyamat-halmazként követel meg beszállítóitól, ez az úgynevezett „HIS-scope”, melyet a 6.8.5 ábra mutat be részletesen. Az ábrából is látható, hogy a jelenlegi „HIS-scope” 5 olyan, beszerzéshez kapcsolódó folyamatot is tartalmaz, amelyek nem részei a korábbi ISO 15504 -5 szabványnak.

Az autóiipari beszállítások sajátosságából fakadóan, az Automotive SPICE kiemelt hangsúlyt fektet a *hardvert és szoftvert tartalmazó rendszerek* fejlesztésére, a beszerzési folyamatok karbantartására, valamint a követelmények kétirányú követhetőségének biztosítására a rendszerek fejlesztésében.

✦ Jelenleg az Automotive SPICE-nak megfelelő felméréseket az iNTACS (Scheme for Education and Certification of Assessors, www.intacs.info) nevű, 2006 márciusában alakult szervezet által akkreditált auditorok végzik. Az Automotive SPICE felméréseket a VDA-QMC (<http://www.vdaqmc.de/>) fogja össze, ez a szervezet akkreditálja az Automotive SPICE-nak megfelelő felméréseket végző szervezeteket is.

Automotive SPICE®

ISO/IEC 15504 and Automotive SPICE®: Process dimension

Management Process Group (MAN) MAN.1 Organizational alignment MAN.2 Organization management A MAN.3 Project management MAN.4 Quality management A MAN.5 Risk management A MAN.6 Measurement	Engineering Process Group (ENG) A ENG.1 Requirements elicitation A ENG.2 System requirements analysis A ENG.3 System architectural design A ENG.4 Software requirements analysis A ENG.5 Software design A ENG.6 Software construction A ENG.7 Software integration A ENG.8 Software testing A ENG.9 System integration A ENG.10 System testing ENG.11 Software installation ENG.12 Software and system maintenance	Supporting Process Group (SUP) A SUP.1 Quality assurance A SUP.2 Verification SUP.3 Validation A SUP.4 Joint review SUP.5 Audit SUP.6 Product evaluation A SUP.7 Documentation A SUP.8 Configuration management A SUP.9 Problem resolution management A SUP.10 Change request management
The Acquisition Process Group (ACQ) ACQ.1 Acquisition preparation ACQ.2 Supplier selection A ACQ.3 Contract agreement A ACQ.4 Supplier monitoring ACQ.5 Customer acceptance A ACQ.11 Technical requirements A ACQ.12 Legal and administrative requirements A ACQ.13 Project requirements A ACQ.14 Request for proposals A ACQ.15 Supplier qualification	Resource & Infrastructure Process Group (RIN) RIN.1 Human resource management RIN.2 Training RIN.3 Knowledge management RIN.4 Infrastructure	Operation Process Group (OPE) OPE.1 Operational use OPE.2 Customer support
Supply Process Group (SPL) A SPL.1 Supplier tendering A SPL.2 Product release SPL.3 Product acceptance support	Process Improvement Process Group PIM.1 Process establishment PIM.2 Process assessment A PIM.3 Process improvement	Reuse Process Group (REU) REU.1 Asset management A REU.2 Reuse program management REU.3 Domain engineering

A Automotive-SPICE new HIS-Scope not included in ISO/IEC IS 15504-5

6.8.5 ábra: Az Automotive SPICE folyamatai, és a HIS által megkövetelt folyamatok. Forrás: HIS – Working Group Assessment, Version V.29, 2007-11-30.

6.8.4.3.3. A CMMI modell

A CMMI® modellt (Capability Maturity Model Integration) a Carnegie Mellon Egyetem Software Engineering Institute-ja fejlesztette (<http://www.sei.cmu.edu>), az Amerikai Védelmi Minisztérium (US. Department of Defense) támogatásával. A modell első változatát 2000-ben tették közzé.

A modell kidolgozásának célja a korábban a szoftverfejlesztésben, rendszerfejlesztésben és termékfejlesztésben leggyakrabban alkalmazott modellek, megközelítések összevonása volt egyetlen modellé, amelyet bármely, szoftverfejlesztéssel (is) foglalkozó szervezet alkalmazhat, a teljes szervezet érettségének és / vagy egyes folyamatai képességének növelésére.

A CMMI® modell a következő szabványokat / modelleket / megközelítéseket integrálja:

- Capability Maturity Model for Software (SW-CMM) v2.0 draft C
- Electronic Industries Alliance Interim Standard (EIA/IS) 731, Systems Engineering Capability Model (SECM)
- Integrated Product Development Capability Maturity Model (IPD-CMM) v0.98
- A modell az ISO/IEC 15504 Technical Report for Software Process Assessment-ben leírt modellel, vagyis a SPICE modellel is kompatibilis.

A fentiekből következik, hogy a CMMI® modell a következő területeken (diszciplínákban) alkalmazható:

- szoftverfejlesztés (SW)
- rendszerszervezés- és fejlesztés (SE)
- integrált termék- és folyamatfejlesztés (IPPD).

A szoftverfejlesztés a CMMI[®] modell kötelezően választandó diszciplínája. A szervezet tevékenységének típusa szerint, ez a diszciplína kiegészíthető a rendszerszervezésre- és fejlesztésre vonatkozóval, és / vagy az integrált termék – és folyamatfejlesztésre vonatkozóval. A SE diszciplína nem tartalmaz további elemeket a SW diszciplínához viszonyítva: a modell ugyanazon követelményeit különbözőképpen kell értelmezni a szoftver- illetve a rendszerfejlesztésre vonatkozóan. Az integrált termék- és folyamatfejlesztés egy *sajátos működési mód* a szervezeten belül: a rendszer / szoftverfejlesztés különálló csapatokban történik, s ezek tevékenységének összehangolása külön figyelmet, energiát igényel. Ha egy szervezet „integráltan” működik, bizonyos plusz-követelményeket is ki kell elégítenie.

A diszciplína kiválasztása után a CMMI[®] modell alkalmazásához a modell lépcsős vagy folytonos megközelítését is ki kell választani.

A lépcsős megközelítés *a szervezet egészére* vonatkozóan határoz meg *érettségi szinteket*. A folytonos megközelítés az *egyes folyamatokra* vonatkozóan *képességi szinteket* azonosít.

2006. augusztus 25-én adta ki a SEI a CMMI[®] v1.2-es változatát, mely 2007 augusztusától teljesen leváltotta az 1.1-es verziót.

A CMMI[®] modell alkalmazásával kapcsolatos tapasztalatok alapján, a CMMI[®]-t fejlesztő csapat több változtatást is eszközölt a modell szerkezetében, a folyamatokban, valamint a sajátos és általános célok között.

A CMMI v1.2 modell leírása a CMMI[®] for Development, Version 1.2, CMMI-DEV, V1.2, CMU/SEI-2006-TR-008, ESC-TR-2006-008. Improving processes for better products. Címen, a <http://www.sei.cmu.edu/cmmi/models/> helyen található.

Az új verzióban a korábbihoz képest eszközölt módosítások célja, lényege a következő:

- a CMMI[®] modell komplexitásának és méretének csökkentése,
- a modell által lefedett területek növelése,
- a modell auditálásával kapcsolatos bizalom erősítése.

A modell módosításával összhangban módosították az auditálásra és oktatásra vonatkozó követelményeket is.

• A CMMI szerkezete

A CMMI[®] v1.2 modell alapja a CMMI[®] Keret (CMMI[®] Framework), mely tartalmazza a modell leírását, a modell auditálásának leírását, valamint a modell oktatásával kapcsolatos anyagokat.

A CMMI[®] modellnek több része van. Egyik a minden nézetben közös un. „Shared CMMI[®] material” (Megosztott CMMI[®] anyag), amely a modell szerkezetének leírását, elemeit, valamint az általánosan használható folyamatokat írja le.

A megosztott (minden esetben érvényes) anyagokon kívül vannak un. „CMMI[®] Constellations”, „CMMI[®] konstellációk”.

A konstellációk tulajdonképpen valamilyen szakterülethez kapcsolódó sajátos folyamatokat írnak le. A következő ilyen szakterületeket azonosították: fejlesztés, beszerzés, szolgáltatás. Az alapelv az, hogy a CMMI[®] modell „megosztott” anyagait

minden esetben használni kell, míg a különböző konstellációkra vonatkozó részeket az egyes szakterületek alkalmazóinak kell használniuk.

Ezidáig (2008 március) a fejlesztésre (CMMI-DEV v1.2, lásd [13]) és a beszerzésre (CMMI-ACQ v1.2, lásd [28]) vonatkozó részek kerültek leírásra. 2008-ban várható a szolgáltatás sajátosságait leíró rész elkészülése.

A CMMI[®] modell legnagyobb előnye, hogy átjárást biztosít a korábban két külön modellben leírt lépcsős és folytonos megközelítések között. A modell egységes, mindkét megközelítésben ugyanazon elemeket tartalmazza, kiküszöbölve ezáltal a terminológiai különbségekből származó kellemetlenségeket. A modellt alkalmazó szervezet vizsgálhatja a teljes szervezet érettségét vagy az egyes folyamatok képességét, de, mivel a folyamatok azonosak, a két megközelítés kölcsönösen megfeleltethető egymásnak.

A CMMI modell mindkét megközelítésében a következő alapelemeket találjuk::

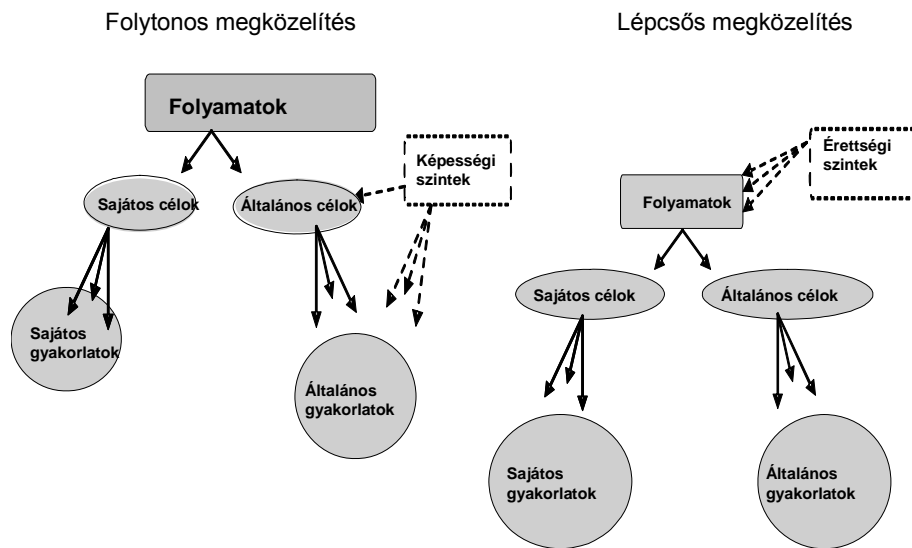
- Folyamatok ⁶(Process area, PA)
- Sajátos célok (Specific goals, SG)
- Sajátos gyakorlat (Specific practices, SP)
- Általános célok (Generic goals, GG)
- Általános gyakorlat (Generic practices, GP)

Lépcsős megközelítésben a folyamatokat érettségi szintek (Maturity Levels, ML) szerint csoportosíthatjuk. Minden érettségi szinthez bizonyos számú, jól meghatározott folyamat tartozik.

Folytonos megközelítésben minden folyamatnak képességi szintjei (Capability Levels, CL) vannak. Ezen felül a folyamatok csoportokba sorolhatók, aszerint, hogy milyen célt szolgálnak a szervezetben.

A CMMI szerkezete a következő:

⁶ A „Process Area” (PA) tulajdonképpen „Folyamat-területet” jelent. Úgy véljük azonban, hogy ez a fordítás magyarul túl bonyolult kifejezéshez vezetne (szó lesz például „PA category” –ról, amely a hasonló célt szolgáló folyamatok csoportját jelenti, s ezt értelmetlen lenne „Folyamat-terület kategória / csoport”-nak fordítani. Emiatt azt a konvenciót alkalmazzuk, hogy a „Process Area” helyett magyarul a „folyamat” elnevezést használjuk.



6.8.6. ábra: A CMMI[®] V1.2 szerkezete

A CMMI[®] modell pontos meghatározást tartalmaz minden elem esetében. Az alábbiakban felsoroljuk a legfontosabb elemekhez kapcsolódó definíciókat.

- Sajátos célok (specific goals - SG): leírják, hogy mit kötelező implementálni ahhoz, hogy kielégítsük az adott folyamat követelményeit.
- Sajátos gyakorlatok (specific practices - SP): a sajátos célok eléréséhez szükséges munkák.
- Általános célok (generic goals - GG): lépcsős megközelítésben minden folyamatsopornak csak egy általános célja van; ez megmutatja, hogy a folyamat mikor eredményes, ismételhető és tartós. Az általános céllal határozzuk meg, hogy egy folyamatsoport mikor kielégített.
- Általános gyakorlatok (generic practices - GP): Intézményesítést nyújtanak, mely biztosítja, hogy a folyamatsoporthoz rendelt folyamatok eredményesek, ismételhetőek és tartósak.
- Alábontott /részletes gyakorlatok (subpractices). A CMMI[®]-ben a gyakorlatokhoz minden esetben kapcsolódnak alábontott gyakorlatok. Ezek részletes leírások, melyek útmutatást nyújtanak a sajátos és általános gyakorlatok értelmezéséhez; alkalmazási ötleteket adnak, melyek hasznosak lehetnek a folyamatfejlesztésben.
- Tipikus munkatermék-leírások (typical work products): a CMMI[®] modell tartalmaz még tipikus munkatermék-leírásokat, amelyek abban segítenek, hogy a CMMI[®]-t alkalmazók megértsék, mi szolgáltathat bizonyítékot arra nézve, hogy egy gyakorlatot ténylegesen végrehajtanak egy szervezetben.
- Szakterülethez kapcsolódó „hatás-erősítők” (discipline amplifications): a modellben leírásra kerültek még ún. „szakterülethez kapcsolódó hatás-erősítők” (discipline amplifications), amelyek a szoftverfejlesztés, rendszerfejlesztés vagy integrált

munkavégzés esetében nyújtanak útmutatást a CMMI® követelmények értelmezésében.

A sajátos és általános célok megértésének elősegítése érdekében, alább példát mutatunk egy CMMI-ben leírt folyamat sajátos és általános céljaira és gyakorlataira.

- Folyamat neve: : konfigurációmenedzsment.
- Folyamat célja: konfigurációmenedzsment célja a munkatermékek integritásának kialakítása és fenntartása konfigurációazonosítás, kontroll, állapot-azonosítás és konfigurációs auditok segítségével.
- Folyamat sajátos céljai és sajátos gyakorlatai (minden folyamat estében egyediek):

SG 1 Alapkonfigurációk létrehozása

SP 1.1 Konfigurációs elemek azonosítása

SP 1.2 Konfigurációmenedzsment rendszer létrehozása

SP 1.3 Alapkonfigurációk létrehozása vagy kibocsátása

SG 2 Változáskövetés- és ellenőrzés

SP 2.1 Változáskérések követése

SP 2.2 Konfigurációs elemek ellenőrzése

SG 3 Integritás biztosítása – részben teljesül

SP 3.1 Konfigurációmenedzsment - feljegyzések készítése

SP 3.2 Konfigurációs auditok végrehajtása

- Folyamat általános céljai és általános gyakorlatai (azonosak az összes többi folyamat esetében is):

GG1: Sajátos célok elérése (Achieve Specific Goals) (gyakorlatilag: a folyamat létezik, végrehajtják, eleget képes tenni sajátos céljainak)

GP1.1. Sajátos gyakorlatok végrehajtása(Perform specific practices)
(végrehajtják a folyamatot)

GG 2 Egy menedzselt folyamat intézményesítése (Institutionalize a Managed Process)

GP 2.1 Szervezeti irányvonal meghatározás (Establish an Organizational Policy)

GP 2.2 A folyamat tervezése (Plan the Process)

GP 2.3 Erőforrás rendelkezésre bocsátása (Provide Resources)

GP 2.4 Felelősség kijelölése (Assign Responsibility)

GP 2.5 Emberek képzése (Train People)

GP 2.6 Konfigurációk menedzsmentje (Manage Configurations)

GP 2.7 Az érintettek azonosítása és bevonása (Identify and Involve Relevant Stakeholders)

GP 2.8 Folyamatkövetés és -vezérlés (Monitor and Control the Process)

GP 2.9 A megfelelés tárgyilagos kiértékelése (Objectively Evaluate Adherence)

GP 2.10 Az aktuális állapot szemlézése a felsőbb vezetőkkel (Review Status with Higher Level Management)

GG 3 Meghatározott folyamat intézményesítése (Institutionalize a Defined Process)

GP 3.1 Meghatározott folyamat létrehozása (Establish a Defined Process)

GP 3.2 Javítási / fejlődési információk összegyűjtése (Collect Improvement Information)

GG 4 Mennyiségileg menedzselt folyamat intézményesítése (Institutionalize a Quantitatively Managed Process)

- GP 4.1 A folyamat mennyiségi mutatóinak meghatározása (Establish Quantitative Objectives for the Process)
- GP 4.2 A folyamat / alfolyamatok teljesítményének stabilizálása (Stabilize Subprocess Performance)
- GG 5 Optimalizáló folyamat intézményesítése (Institutionalize an Optimizing Process)
 - GP 5.1 Folyamatos folyamatjavítás biztosítása (Ensure Continuous Process Improvement)
 - GP 5.2 A problémák okának (gyökerének) korrekciója (Correct Root Causes of Problems)

- **CMMI DEV1.2**

A CMMI® megosztott anyagaiban leírt (vagyis minden esetben alkalmazandó) folyamatok a következők (az angol rövidítések betűrendjében):

- CAR – Causal Analysis and Resolution, Oksági elemzés és megoldás
- CM – Configuration Management, Konfigurációmenedzsment
- DAR – Decision Analysis and Resolution, Döntéselemzés és közzététel
- IPM – Integrated Project Management, Integrált projektmenedzsment
- MA – measurement and Analysis, Mérés és elemzés
- OID – Organizational Innovation and Deployment, Szervezeti szintű innováció és közzététel
- OPD – Organizational Process Definition, Szervezeti szintű folyamatok meghatározása
- OPF – Organizational Process Focus, Szervezeti szintű folyamatszemplélet
- OPP – Organizational Process Performance, Szervezeti szintű folyamat teljesítmény
- OT – Organizational Training, Szervezeti szintű képzés
- PMC – Project Monitoring and Control, Projektkövetés és –vezérlés
- PP – Project Planning, Projekttervezés
- PPQA – Product and Process Quality Assurance, Termék és folyamat minőségbiztosítás
- QPM – Quantitative Project Management, Mennyiségi projektmenedzsment
- REQM – Requirements Management, Követelménymenedzsment
- RSKM – Risk Management, Kockázatmenedzsment.

A fejlesztéshez kapcsolódó (tehát a fejlesztési konstellációban leírt) folyamatok a következők: (szintén az angol rövidítés betűrendjében):

- PI – Product Integration, Termék integráció
- RD – Requirements Development, Követelményfejlesztés
- SAM – Supplier Agreement Management, Beszállítói megállapodás menedzsment
- TS – Technical Solution, Műszaki megoldás
- VAL – Validation, Validáció
- VER – Verification, Verifikáció.

A folyamatokat, és a lépcsős nézetben alkalmazott, hozzájuk rendelt érettségi szintet a következő táblázat tartalmazza.

Folyamat, magyarul	Folyamat, angolul	Rövidítés	ML	CL1	CL2	CL3	CL4	CL5
Követelménymenedzsment	Requirements management	REQM	2	2-es szint célprofilja				
Mérés és elemzés	Measurement and analysis	MA	2					
Projektkövetés és vezérlés	Project monitoring and control	PMC	2					
Projekt tervezés	Project planning	PP	2					
Folyamat és termék minőségbiztosítás	Process and product quality assurance	PPQA	2					
Beszállítói megállapodás menedzsment	Supplier agreement management	SAM	2					
Konfigurációmenedzsment	Configuration management	CM	2					
Döntéselemzés és közzététel	Decision analysis and resolution	DAR	3	3-as szint célprofilja				
Termék integráció	Product integration	PI	3					
Követelményfejlesztés	Requirements development	RD	3					
Műszaki megoldás	Technical solution	TS	3					
Validáció	Validation	VAL	3					
Verifikáció	Verification	VER	3					
Szervezeti szintű folyamatok meghatározása + IPPD	Organizational process definition + IPPD	OPD + IPPD	3					
Szervezeti szintű folyamatszemplét	Organizational process focus	OPF	3					
Integrált projektmenedzsment + IPPD	Integrated project management + IPPD	IPM + IPPD	3					
Kockázatmenedzsment	Risk management	RSKM	3					
Szervezeti szintű képzés	Organizational training	OT	3					
Szervezeti szintű folyamat teljesítmény	Organizational process performance	OPP	4	4-es szint célprofilja				
Mennyiségi projekt menedzsment	Quantitative project management	QPM	4					
Szervezeti szintű innováció és annak bevezetése	Organizational innovation and deployment	OID	5	5-ös szint célprofilja				
Oksági elemzés és megoldás	Causal analysis and resolution	CAR	5					

6.8.1. Táblázat A szervezet érettsége és a folyamatok képessége közötti kapcsolat a CMMI®-DEV v1.2 modellben

A táblázatból látható, hogy egy cég akkor van 2-es érettségi szinten, ha a Követelménymenedzsment, Mérés és elemzés, Projekt tervezés, Projektkövetés és vezérlés, Folyamat és termék minőségbiztosítás, Beszállítói megállapodás menedzsment és Konfigurációmenedzsment folyamatokat üzemelteti, és ezek képességi szintje legalább 2-es. A szervezet 3-as érettségi szintjéhez további 11 folyamat szükséges, és az, hogy mind a 2-es, mind a 3-as érettségi szinthez tartozó folyamatok képessége legalább 3-as legyen. 4-es és 5-ös érettségi szinten további két-két folyamat jelenléte szükséges, de elegendő ezek 3-as képességi szintje.

Ebből az látható, hogy a lépcsős megközelítés egy, a cég egészére vonatkozó, elfogadható (3-as) képességi szintű folyamatok üzemeltetését teszi szükségessé. Egyes folyamatok képességét folytonos megközelítést alkalmazva lehet 3-as fölé fejleszteni.

- **CMMI-ACQ v1.2.**

A beszerzéshez kapcsolódó (tehát a CMMI-ACQ konstellációban leírt), megosztott anyagokban nem szereplő folyamatok a következők: (szintén az angol rövidítés betűrendjében):

- AM - Agreement Management, Megállapodás-menedzsment
- ARD - Acquisition Requirements Development, Beszerzéshez kapcsolódó követelmények fejlesztése
- ATM - Acquisition Technical Management , Beszerzéshez kapcsolódó műszaki menedzsment
- AVAL - Acquisition Validation, Beszerzéshez kapcsolódó validáció
- AVER - Acquisition Verification, Beszerzéshez kapcsolódó verifikáció
- SSAD - Solicitation and Supplier Agreement Development, Kérelmezés és beszállítói megállapodások fejlesztése

A többi folyamat azonos a CMMI megosztott anyagaiban szereplő folyamatokkal, s az azonosság igaz a CMMI-ACQ v1.2 szerkezetére, elemeinek kapcsolatára is. Az alábbi táblázatban megfigyelhető, melyik érettségi szinthez kapcsolták a CMMI-ACQ konstelláció folyamatait.

Folyamat, magyarul	Folyamat, angolul	Rövidít és	ML	CL1	CL2	CL3	CL4	CL5
Megállapodás menedzsment	Agreement Management	AM	2	2-es szint cél- profilja				
Beszerezéshez kapcsolódó követelmények fejlesztése	Acquisition Requirements Development	ARD	2					
Konfigurációmenedzsment	Configuration Management	CM	2					
Mérés és elemzés	Measurement and Analysis	MA	2					
Projektkövetés és – vezérlés	Project Monitoring and Control	PMC	2					
Projekttervezés	Project Planning	PP	2					
Folyamat- és termékminőség biztosítás	Process and Product Quality Assurance	PPQA	2					
Követelménymenedzsment	Requirements Management	REQM	2					
Kérelmezés és beszállítói megállapodások fejlesztése	Solicitation and Supplier Agreement Development	SSAD	2					
Beszerezéshez kapcsolódó műszaki menedzsment	Acquisition Technical Management	ATM	3	3-as szint célprofilja				
Beszerezéshez kapcsolódó validáció	Acquisition Validation	AVAL	3					
Beszerezéshez kapcsolódó verifikáció	Acquisition Verification	AVER	3					
Döntésemelés és közzététel	Decision Analysis and Resolution	DAR	3					
Integrált projektmenedzsment	Integrated Project Management	IPM	3					
Szervezeti szintű folyamatok meghatározása	Organizational Process Definition	OPD	3					
Szervezeti szintű folyamatszemplét	Organizational Process Focus	OPF	3					
Szervezeti szintű képzés	Organizational Training	OT	3					
Kockázatmenedzsment	Risk Management	RSKM	3					
Szervezeti szintű folyamat-teljesítmény	Organizational Process Performance	OPP	4	4-es szint célprofilja				
Mennyiségi projektmenedzsment	Quantitative Project Management	QPM	4					
Oksági elemzés és megoldás	Causal Analysis and Resolution	CAR	5	5-ös szint célprofilja				
Szervezeti szintű innováció és annak közzététele	Organizational Innovation and Deployment	OID	5					

2. Táblázat A szervezet érettsége és a folyamatok képessége közötti kapcsolat a CMMI®-ACQ v1.2 modellben

A 2. táblázatból látható, hogy az ACQ konstellációra jellemző folyamatok többsége a 2-es érettségi szinthez kapcsolódik. Ezek a folyamatok a beszállítóval való kapcsolattartásra, a beszállítói megállapodások elkészítésére és menedzselésére vonatkoznak. 3-as érettségi szinten a műszaki folyamatok hasonlóak a fejlesztésben jelen lévőkhöz, de itt is hangsúlyos, hogy ezek mindegyike a beszállító műszaki folyamatira (követelményfejlesztés, tervezés, tesztelés) vonatkoznak. A 4-es és 5-ös érettségi szinten nincs különbség a CMMI-DEV és a CMMI-ACQ folyamatai között.

- A CMMI modell alapján végzett felmérések

A CMMI[®] modellnek való megfelelés felmérésre vonatkozó követelmények a CMMI[®] keret részei. A felmérés általános követelményeit az „Appraisal Requirements for CMMI[®]” (röviden ARC, lásd [29]) tartalmazza. Az ARC alapján bármely szervezet kidolgozhatja saját módszertanát, mely alkalmas arra, hogy a CMMI[®] modellnek való megfelelést ellenőrizze.

Ennek az ARC követelményrendszernek az alapján a SEI kidolgozta saját felmérési módszertanát, a SCAMPISM-t („Standard CMMI[®] Appraisal Method for Process Improvement”), melynek 3 osztályát különböztette meg (lásd [30], [31]): SCAMPISM A, SCAMPISM B és SCAMPISM C osztályt. A SCAMPISM módszertan alkalmazásával végzett felméréseket a SEI tartja kézben, figyeli és követi őket, és nemzetközi adatbázist működtet a felmérések adatainak feljegyzésére. Az ilyen felméréseket csak a SEI által képzett és felhatalmazott SCAMPISM vezető auditorok (SCAMPISM Lead Appraiser) végezhetik. A felmérésekkel kapcsolatos információkat a vezető auditor jelenti a SEI-nek. Ilyenformán, a világban végzett összes SCAMPISM felmérés összefoglaló adatai a SEI rendelkezésére állnak.

felmérésekkel kapcsolatos információk a <http://www.sei.cmu.edu/cmmi/appraisals/appraisals.html> helyen, a SEI által nyilvántartott SCAMPISM vezető auditorok listája a <http://www.sei.cmu.edu/appraisal-program/directory/scampi-appraisers.html> címen érhető el.

A CMMI[®] modellnek való megfelelést ellenőrző felmérésekre vonatkozó követelmények a CMMI[®] modell v1.2-es verziójának megjelenésekor, 2006 nyarán szintén frissültek. Jelenleg az ARC-nek és a SCAMPISM A módszertannak a v1.2 verziója érvényes. 2005-ben tették közzé a SCAMPISM B és C módszertanokat. Mivel korábban nem volt B és C típusú felmérést leíró módszertan, ez az első verzió, amelyet V1.1-es verziószámmal jelöltek.

A SCAMPISM módszertan szerint végzett audit során felmérik, hogy az adott cég vagy folyamatcsoport megfelel-e a célként kitűzött érettségi vagy képességi szintnek. A felmérés során tulajdonképpen azt vizsgálják, hogy az egyes folyamatok mennyire teljesítik a rájuk vonatkozó sajátos és általános célokat. A célok teljesülésének vizsgálatához ún. „objektív bizonyítékokat” (objective evidence) gyűjtenek, amelyeknek több forrásból kell származniuk. Eljárásokból, dokumentumokból, valamint a szervezet tagjainak szóbeli beszámolójából származó objektív bizonyítékot kell beszerezni minden vizsgált folyamat minden sajátos és általános gyakorlatára vonatkozóan. A felmérés során a szervezet egészét, vagy annak kijelölt részeit (pl. egy osztályát) vizsgálják. A tanúsítványt a vizsgált terület fogja megkapni. A kijelölt szervezeti egységen belül bizonyos számú (reprezentatív mintának számító) projektben is vizsgálni kell a folyamatok végrehajtását. A SCAMPISM A felmérés egy tervezési fázissal kezdődik, majd az objektív bizonyítékok összegyűjtése következik. Ebben a vezető auditort a felmérést végző csapat tagjai segítik.

A felmérést végző csapat tagjai lehetnek külső szervezet tagjai, de ajánlatos, hogy a felmért szervezet is delegáljon tagokat a felmérést végző csapatba, mert az objektív bizonyítékok összegyűjtése, rendszerezése és dokumentálása hosszú folyamat, melyet lényegesen lerövidíthet, ha az auditorok egy része a felmért szervezet tagja, s jól ismeri a szervezetben alkalmazott eljárásokat, a dokumentumok leelőhelyét stb..

A felmérést végző csapat összetételére, a tagok szakképesítésére pontos előírásokat tartalmaz a SCAMPISM A módszertan. Követelmény például, hogy minden csapattagnak (már a felmérés előtt) részt kell vennie a SEI által szervezett hivatalos, 3 napos „CMMI®-intro” képzésen. Ezen kívül, a vezető auditor a felmérés megkezdésekor ismét rövid képzést tart a csapatnak a felmérés menetéről.

Az objektív bizonyítékok dokumentálását követően kezdődhet a helyszíni szemle, amely általában 1-2 hétig tart (a vizsgált folyamatok számától függően). Az auditorok ekkor már nem „felfedezik”, hanem csak ellenőrzik az objektív bizonyítékokat. Jól meghatározott lefedettségű követelmények teljesülését is vizsgálni kell, majd, ha a felmérés végző csapat úgy ítéli meg, hogy az összegyűjtött objektív bizonyítékok száma és minősége megfelelő, következik a célok teljesülésének vizsgálata, bizonyos, jól meghatározott algoritmus szerint. A célok teljesüléséről való döntés csapatszintű, azaz a felmérést végző csapat minden tagjának el kell fogadnia minden döntést.

Egy folyamat akkor tekinthető megfelelőnek, ha minden sajátos célja, valamint az adott érettségi / képességi szinten megkövetelt minden általános célja teljesül.

A felmérés eredményéből összefoglaló születik, melyet a felmért szervezetnek bemutatnak. A felmérés összefoglaló jelentését elküldik a SEI-nek. A továbbiakban, a SEI bármikor ellenőrizheti a felmérés bármely anyagát, és a felmért szervezetnél is végezhet ellenőrzést, ha valami rendellenességet tapasztal.

A SCAMPISM B, illetve C típusú felmérések is objektív bizonyítékok összegyűjtésével vizsgálják a sajátos és általános gyakorlatokat. Különbség a szükséges bizonyítékok számában van (kevesebb számú és fajtájú objektív bizonyíték is elegendő), valamint abban, hogy SCAMPISM B és C típusú felméréseknél az értékelés legmagasabb megengedett szintje a gyakorlat-szintű értékelés. Nem megengedett célok teljesülését, folyamatok képességi szintjét, vagy a szervezet érettségi szintjét vizsgálni. Ez a megszorítás logikus, hiszen a kevesebb számú és típusú objektív bizonyíték csak korlátozott rálátást tesz lehetővé a szervezetre. A SCAMPISM B és C típusú felmérések eredményeképpen szöveges értékelés is születik, amely a folyamatfejlesztésre vonatkozó tanácsokat is tartalmaz.

A összefoglaljuk az A, B és C típusú a felmérések néhány követelményét.

Követelmények	A osztályba tartozó audit	B osztályba tartozó audit	C osztályba tartozó audit
Szükséges objektív bizonyítékok típusa	Dokumentumok és interjúk	Dokumentumok és interjúk	Dokumentumok vagy interjúk
Értékelés	Célok értékelése szükséges	Nem megengedett	Nem megengedett
Szervezet lefedettsége	Szükséges	Nem szükséges	Nem szükséges
Felmérést végző csapat minimális létszáma	4	2	1
A felmérést végző csapat vezetőjére	Vezető auditor	Képzett és tapasztalt személy	Képzett és tapasztalt személy

vonatkozó
követelmény

6.8.3 Táblázat: A különböző típusú CMMI[®] auditok követelményei (Forrás: ARC 1.2)

A SCAMPISM v1.2 módszertan bevezette azt a megszorítást, hogy a felmérések csak 3 évig érvényesek.

2007 januárjától már kizárólag az ARC1.2 (SCAMPISM 1.2) alapján lehet felméréseket végezni. 2007 augusztusáig még lehetőség volt a felméréseken a CMMI[®] v1.1 modell követelményeit ellenőrizni. 2007 augusztusától csak a CMMI[®] v1.2 verziójú modellt lehet alkalmazni.

6.8.4.4. Folyamatfejlesztési modellek összekapcsolása

A világ vezető folyamatfejlesztéssel foglalkozó intézeteiben több kísérlet is történt és történik arra vonatkozóan, hogy különböző folyamatfejlesztési modelleket összekapcsoló, integrált megközelítések alakuljanak ki. Már kezdetben próbálták az általános ISO 9001-et összekapcsolni a szoftverfejlesztést támogató ISO 9000-3-mal, így született meg az elsősorban Nagy Britanniában nagy népszerűsége szert tett TickIT módszertan.

Az ESI (European Software Institute - www.esi.es) fejlesztette ki a SPICE- EFQM kombinációjából létrejött tanúsítható modellt.. Szintén az ESI fejlesztette az ITMark nevű megközelítést, amely tulajdonképpen kis cégek számára kínál egy CMMI[®]-szerű megközelítést és tanúsítást (lásd <http://www.esi.es>) .

Helyi kis kezdeményezések (pl ESI-nek), országos próbálkozások voltak még az ISO 9001 és a CMM, a CMM és SPICE, valamint a CMMI[®] és ISO 9001 kombinációira. Ezek inkább kutatási témaként jelentek meg, vagy adott konkrét helyzetben alkalmazták őket (pl. egy cégnek szüksége volt egy SPICE és EFQM tanúsításra) és nem kerültek be a mindennapi gyakorlatba.