

28 A FOLYAMAT TOVÁBBFEJLESZTÉSE

TÉMAKÖRÖK

A fejezet bemutatja, hogyan lehet továbbfejleszteni a szoftverfolyamatokat jobb szoftverek előállítása érdekében. A fejezet elolvasása után:

- megértjük a szoftverfolyamat továbbfejlesztésének alapjait és azt, hogy miért érdemes foglalkozni a folyamat továbbfejlesztésével;
- megértjük a szoftverfolyamat tényezőinek a szoftver minőségére és a szoftverfejlesztők termelékenységére gyakorolt hatását;
- képesek leszünk a szoftverfolyamat egy egyszerű modelljének ki-fejlesztésére;
- megértjük a folyamatképesség és folyamatfejlettség fogalmát, és a folyamat továbbfejlesztésére szolgáló CMMI-modell általános alakját.

TARTALOM

- 28.1. A folyamat és a termék minősége
- 28.2. A folyamat osztályozása
- 28.3. A folyamat mérése
- 28.4. A folyamat elemzése és modellezése
- 28.5. A folyamat megváltoztatása
- 28.6. A CMMI folyamat-továbbfejlesztő keretrendszer

Ahogy azt a 27. fejezetben említettem, a fejlesztés alatt álló szoftvertermék minősége és a termék előállításánál alkalmazott szoftverfolyamat minősége között szoros összefüggés áll fenn: a szoftverfolyamat továbbfejlesztésével együtt remélhetően a kapcsolódó termék minősége is hasonlóképpen növekszik. A *folyamat továbbfejlesztése* a meglévő folyamatok megértését és megváltoztatását jelenti, célja pedig a termék minőségének javítása és/vagy a költségek és a fejlesztési idő csökkentése. Az ezzel foglalkozó szakirodalom nagyobbik részében a folyamat továbbfejlesztését a termék minőségének növelése és főleg az átadott szoftver hibaszámának csökkentése szempontjából tárgyalják. E célok megvalósítása után a költségek és az ütemezés csökkentése válhatnak az elsődleges célokká.

A szoftverfolyamatok eredendően összetettek és nagyon sok tevékenységet fognak össze, és ugyanúgy, mint a termékek, szintén rendelkeznek jellemzőkkel (28.1. ábra). Lehetetlen a folyamat olyan továbbfejlesztését végrehajtani, amely a folyamat összes jellemzőjét egyidejűleg optimalizálja. Ha például gyors fejlesztési folyamatra van szükség, akkor elkerülhetlenné válhat a láthatóság csökkentése, de mivel a folyamat láthatóvá tétele alatt dokumentumok rendszeres időközönként való előállítása értendő, ezért a láthatóság lassítaná a folyamatot.

Folyamatjellemzők	Leírás
Érthetőség	Mennyire pontosan definiálták a folyamatot, és milyen könnyű megérteni ezt a definíciót.
Láthatóság	Világosak-e, hogy mit eredményeznek a folyamat tevékenységei, és így kívülről is látható-e a folyamat haladása.
Támogathatóság	Milyen mértékben támogatják a CASE-eszközök a folyamat tevékenységeit.
Elfogadhatóság	A definiált folyamat elfogadható-e és használható-e a szoftvertermék elkészítéséért felelős mérnökök számára.
Megbízhatóság	Úgy tervezték-e meg a folyamatot, hogy a folyamat hibái kiküszöbölhetők vagy megszüntethetők legyenek, mielőtt a szoftvertermék elkészülne.
Stabilitás	Lehet-e váratlan problémák felmerülése esetén is folytatni a folyamatot.
Karbantarthatóság	A szervezeti követelmények változásaira vagy az azonosított folyamat-továbbfejlesztésekre reagálva megváltoztatható-e a folyamat.
Sebesség	A folyamat alapján milyen gyorsan lehet adott specifikációból szállítható rendszert előállítani.

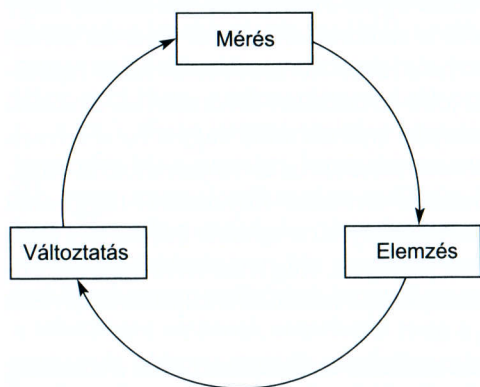
28.1. ábra. A folyamatjellemzők

A folyamat továbbfejlesztése nem egyszerűen azt jelenti, hogy bizonyos módszereket, eszközöket vagy egy folyamat valamilyen máshol már felhasznált modelljét kell alkalmazni. Bár az ugyanolyan típusú szoftvert fejlesztő szervezeteknek kétségtelenül sok közös vonásuk van, mindig vannak olyan helyi szervezeti tényezők, folyamatok és szabványok, amelyek befolyásolják a folyamatot, ezért nem valószínű, hogy a másutt már használt folyamat-továbbfejlesztés egyszerű bevezetése sikerrel jár. A folyamat-továbbfejlesztést mindig egy szervezetre, vagy nagyobb szervezetek esetében a szervezet részeire nézve egyedi tevékenységként kell kezelni.

A folyamat-továbbfejlesztés ciklikus tevékenység, ahogy azt a 28.2. ábra szemlélteti. Három fő lépése van:

1. *Folyamatmérés.* Az aktuális projektet vagy a terméket mérjük. A cél a szervezet célkitűzéseivel (beleértve a folyamat továbbfejlesztését is) összhangban a mérések pontosítása.

2. *Folyamatelemzés.* Felmérjük a folyamatot, azonosítjuk a gyengeségeket és a szűk keresztmetszeteket. A folyamatot leíró folyamatmodellt szokásosan ebben a lépésben fejlesztjük ki.
3. *Folyamatváltoztatás.* Az elemzés alatt feltárt folyamatváltoztatások bevezetésre kerülnek.



28.2. ábra. A folyamat továbbfejlesztési ciklusa

Ezeket a tevékenységeket a fejezetben később külön alfejezetekben tárgyalom. A folyamat lépései akár több hónapig is eltarthatnak; a folyamat továbbfejlesztése hosszú ideig tartó tevékenység. Ugyanakkor folytonos tevékenység is, akár új folyamat kerül bevezetésre, akár az üzleti környezet változik, és ezek a folyamatok maguk úgy fognak fejlődni, hogy a változtatások megtörténnek.

28.1. A FOLYAMAT ÉS A TERMÉK MINŐSÉGE

A folyamat továbbfejlesztésének kiindulópontja az a feltevés, hogy a termék minőségét befolyásoló kritikus tényező nem más, mint a termékfejlesztési folyamat minősége. A folyamat továbbfejlesztésének ötlete W. E. Deming amerikai mérnök munkájából származik, aki a II. világháború után minőségjavítással foglalkozott egy japán gyárban. A japán ipar sok éve a folyamat folyamatos továbbfejlesztése mellett kötelezte el magát, ami nagyban hozzájárult a Japánban gyártott termékek általános magas minőségéhez.

Deming (és mások) vezették be a statisztikai minőség-ellenőrzés gondolatát, amely a termék hibaszámának mérésén, illetve a hibák és a folyamat közötti összefüggés vizsgálatán alapszik. A folyamat továbbfejlesztésének célja a termék hibaszámának csökkentése, és a továbbfejlesztését addig ismételjük, amíg lehet, vagyis ameddig a folyamat eredménye előre jelezhető és csökken a hibák száma. Ekkor szabványosítják a folyamatot, és újabb javítási ciklus kezdődik.

Humphrey folyamatkezelést tárgyaló alapkönyvében (Humphrey, 1988) megjegyzi, hogy ugyanezek a technikák alkalmazhatók a szoftverfejlesztésben. Azt állítja:

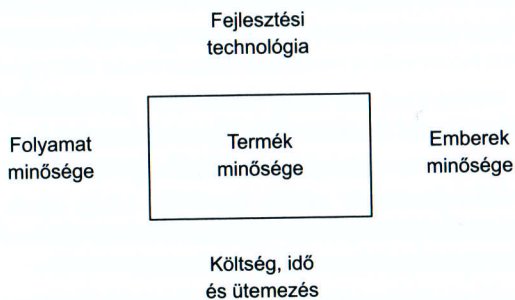
W. E. Deming a II. világháború utáni japán gyárakról szóló munkájában a statisztikai folyamat-ellenőrzés elvét alkalmazta a gyárakra. Bár jelentős különbségek vannak, ezek az elvek ugyanúgy alkalmazhatók a szoftverre, mint az autóra, kamerára, karóra vagy acélra.

Bár a hasonlóságok világosak, nem értek egyet Humphrey-vel abban, hogy a gyártmánytervezés közvetlenül átvihető a szoftvertervezésre. Gyártás esetén kézenfekvő a folyamat–termék kapcsolat: ha a hibák kiküszöbölése miatt fejlesztik tovább a folyamatot, akkor a folyamat jobb termékeket eredményez. Szellemi termékek esetén a kapcsolat már nem ennyire nyilvánvaló, vagyis ha a termék bizonyos mértékben a nem automatizálható intellektuális folyamatoktól függ. A szoftver minősége sem a gyártási, hanem a tervezési folyamattól függ, ahol jelentős az emberi tényező. Néhány termékosztálynál a termék minőségét elsősorban befolyásoló tényező az alkalmazott folyamat, míg más termékeknél, különösen az innovatív alkalmazásoknál, fontosabbak lehetnek a folyamatban részt vevő emberek, mint maga a folyamat.

A szoftvertermékeknel (vagy más olyan szellemi termékeknel, ahol a minőség elsősorban a tervezéstől függ, például a könyveknel, filmeknel stb.) a termékminőséget a 28.3. ábrán látható négy tényező befolyásolhatja.

Az egyes tényezők hatása a projekt méretétől és típusától függ. A különböző csapatok által fejlesztett független alrendszerekből álló nagyon nagy rendszereknel a szoftverfolyamat legfőbb meghatározója a termék minősége. A nagy projektek fő problémáit az integráció, a projektvezetés és a kommunikáció okozzák. A csapat rendszerint vegyes képességű és tapasztalatú emberekből áll. Mivel a fejlesztési folyamat általában több éven keresztül tart, a fejlesztőcsapat megváltozhat, akár teljesen ki is cserélődhet a projekt élettartama alatt, így a projekt során a nagyobb szaktudású vagy több tapasztalattal rendelkező egyének hatása általában nem domináns.

Ezzel szemben a kicsi és alacsony létszámú projekteknél fontosabb a fejlesztőcsapat minősége, mint a fejlesztési folyamat. Ha a csapat jó képességű és tapasztalt, akkor valószínűleg a termék is jobb minőségű lesz, viszont ha a csapat tapasztalatlan és szakmailag sem felkészült, akkor a jó folyamat ugyan ellensúlyozhatja ezt a veszteséget, de önmagában nem biztosíték a szoftver magas minőségére.



28.3. ábra. A szoftvertermékek legfontosabb minőségi tényezői

Kis csapatok esetében különösen fontos a jó fejlesztési technológia. A csapat nem szánnak sok időt az egyhangú adminisztratív eljárásokra: a mérnökök idejük túlnyomó részét a rendszer tervezésére és programozására fordítják, ezért a jó eszközöknek jelentős hatása lehet a termelékenységre. A nagy projekteknél az információkezeléshez elengedhetetlen az alapszintű fejlesztési technológia, és meglepő módon kevésbé fontosak a CASE-eszközök. A csapattagok idejüknek viszonylag kisebb részét fordítják tevékenységek fejlesztésére, több időt töltenek kommunikációval és a rendszer más részeinek megértésével. Termelékenységük főleg ettől függ, a fejlesztői eszközök ezen nem változtatnak semmit.

A 28.3. ábrán látható téglalap aljánál elhelyezkedő tényezők abszolút kritikus tényezők. Ha a projekt – bármekkora méretű legyen is – költségvetését alultervezik, vagy megvalósíthatatlan szállítási ütemtervet készítenek hozzá, ez hatással lesz a termék minőségére. A jó folyamatok hatékony implementálásához erőforrásokra is szükség van. Ha az erőforrások nem elegendők, akkor a folyamat nem tud hatékonyan működni, ha pedig nem megfelelők az erőforrások, akkor csak a tehetséges emberek menthetik meg a projektet, de ha a költségvetési hiány túl nagy volt, akkor a termék minősége mindenféleképpen romlik.

A szoftverek minőségi problémáinak valódi oka nagyon gyakran nem a gyenge vezetés, a nem megfelelő folyamatok vagy a rossz minőségű betanítás, hanem az a tény, hogy a szervezetnek a túlélésért kell küzdenie. Sok szoftverprojekttnél szándékosan szabják alacsonyra a költségvetést, hogy fejlesztési szerződéshez jussanak. Nem meglepő, hogy az ilyen rendszerekben nehéz biztosítani a termék minőségét.

28.2. A FOLYAMAT OSZTÁLYOZÁSA

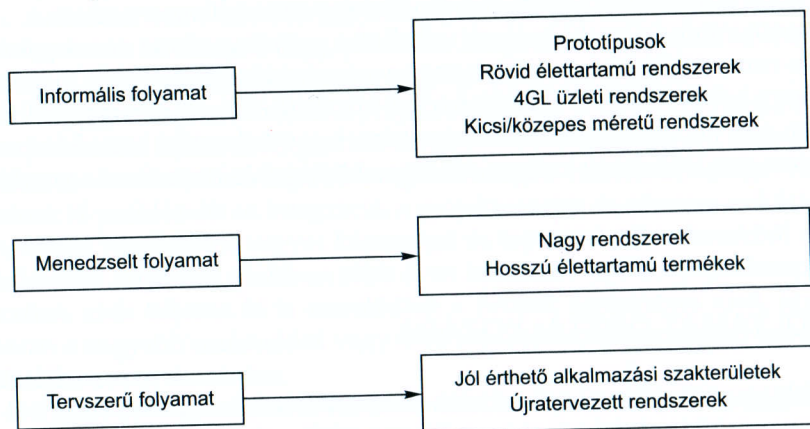
A szoftverfolyamat megfigyelhető minden szervezetben, az egyszemélyes vállalatától a nagy multikig. A folyamat formális szintjétől, a fejlesztett termék típusától, a szervezet méretétől stb. függően ezek a folyamatok különböző típusúak lehetnek. A szoftverfolyamatok négy osztályba sorolhatók:

1. *Informális folyamatok.* Azok a folyamatok, amelyekhez nincsen szigorúan definiált folyamatmodell, és a használni kívánt folyamatot a csapat választja ki. Az informális folyamatok használhatnak olyan formális eljárásokat, mint például a konfigurációkezelés, de az alkalmazandó eljárások és az eljárások közötti kapcsolat nincs előre definiálva.
2. *Menedzselt folyamatok.* Azok a folyamatok, amelyek a fejlesztési folyamat irányítására definiált folyamatmodellt használnak. A folyamatmodell meghatározza a használt eljárásokat, az ütemezésüket és az eljárások közötti kapcsolatokat.
3. *Tervezhető folyamatok.* Azok a folyamatok, amelyekhez valamilyen meghatározott fejlesztési módszert vagy módszereket (például objektumorientált tervezéshez szisztematikus módszereket) alkalmaznak. Ezek a folyamatok kihasználják a CASE-eszközök tervezéshez és elemzéshez nyújtott támogatását.

4. *Tökéletesítő folyamatok.* Az eredendően fejlesztési célokat tartalmazó folyamatok. A folyamat továbbfejlesztésére és az ilyen fejlesztések bevezetésére működtetett eljárásokról külön költségvetésben gondoskodnak. A fejlesztések részeként akár mennyiségi folyamatmérések is bevezethetők.

Ezek az osztályok nyilvánvalóan átfedik egymást, és egy folyamat több osztályba is tartozhat. Ha például a folyamatot a fejlesztőcsapat választja ki, akkor az tartozhat az informális folyamatok közé. A csapat dönthet egy konkrét tervezési módszer használatáról, és akár a folyamat továbbfejlesztése is lehetséges, ekkor a folyamat *informális, tervszerű és tökéletesítő* folyamat lesz.

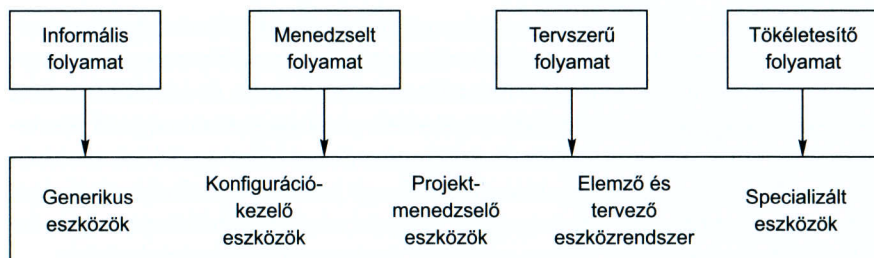
Az osztályozások hasznosak, mert alapot adnak a többdimenziós folyamat-továbbfejlesztéshez, és segítik a szervezetet a termékfejlesztés különböző típusaihoz megfelelő folyamat kiválasztásában. A különböző terméktípusok és a kifejlesztésükhöz használható folyamat típusok a 28.4. ábrán láthatók. A tökéletesítő folyamatok nem láthatók az ábrán, mivel bármely típusú folyamat lehet tökéletesítő folyamat.



28.4. ábra. Folyamat alkalmazhatósága

A 28.4. ábrán látható osztályok átfedhetik egymást, ezért az újratervezett kis rendszerek kifejleszthetők tervszerű folyamattal. Nagy rendszereknél mindig menedzselt folyamatokra van szükség, viszont ha a szakterületet nem értik meg jól, akkor nehézségekbe ütközhet a megfelelő módszer kiválasztása. Nagy rendszerek kifejlesztésére ezért olyan menedzselt folyamatok használhatók fel, amelyek egyetlen konkrét tervezési módszeren sem alapszanak.

A folyamatosztályozás segítségével kiválasztható egy meghatározott típusú termék kifejlesztéséhez használt megfelelő folyamat. Vegyük például azt az esetet, amikor egy számítógéprendszer-típusok közötti átmenetet támogató programra van szükség. A program élettartama viszonylag rövid, kifejlesztéséhez nincs szükség azokra a szabványokra és vezetési eljárásokra, amelyek az évekig használatos szoftverekhez kellenek.



28.5. ábra. Folyamat eszközökkel való támogatottsága

A folyamatosztályozás felismeri, hogy a folyamat befolyásolja a termék minőségét, de nem feltételezi, hogy mindig a folyamat az uralkodó tényező. Alapot ad a különböző típusú folyamatok továbbfejlesztéséhez, és különböző típusú folyamatokhoz különböző típusú továbbfejlesztés alkalmazható. A tervszerű folyamatok továbbfejlesztése például alapulhat a módszerben megszerzett jobb gyakorlaton, a követelmények és a tervezés jobb összehangolásán, a tökéletesített CASE-eszközökön stb.

Ma már a legtöbb szoftverfolyamat támogatására létezik valamilyen CASE-eszköz, ezért ezek *támogatott folyamatok*. A tervszerű folyamatokat ma általában elemzési és tervezési eszközrendszerek támogatják, viszont a folyamatoknak lehetnek másféle támogatóeszközeik (például prototípus-készítési eszközök, tesztelőeszközök), függetlenül attól, hogy használnak-e strukturált tervezési módszertant vagy sem.

A támogatóeszköz eredményessége a folyamat osztályozásától függ. Az informális folyamatokhoz például használhatók olyan általános eszközök, mint például a prototípus-készítő nyelvek, a fordítóprogramok, a hibakeresők vagy a szövegszerkesztők stb. A 28.5. ábrán a szoftverfejlesztésben használható különböző eszközök egész sora látható. Az egyes eszközök hatékonysága az alkalmazott folyamat típusától függ.

Az elemzési és tervezési CASE-eszközök költséghatékonysága csak ott valószínű, ahol követik a tervszerű folyamatot. A folyamat-továbbfejlesztési tevékenység részeként specializált eszközöket fejlesztettek ki bizonyos folyamattevékenységek fejlesztésének külön támogatására. Például egy több honlapot kezelő fejlesztést végző csapat létrehozhat egy olyan display eszközt, amely egyszerre mutatja az összes honlapon folyó munkát.

28.3. A FOLYAMAT MÉRÉSE

Folyamatméréseknek a szoftverfolyamatból származó mennyiségi adatokat nevezzük. Humphrey (Humphrey, 1989) a folyamat továbbfejlesztéséről szóló könyvében vitatja, hogy a folyamatmetrika-gyűjtemények lényegesek lennének a folyamatok továbbfejlesztésében, de szerinte a mérés fontos szerepet játszik a kis-

méretű, egyéni folyamat-továbbfejlesztésben (Humphrey, 1995). A folyamatmetrikák használhatók annak megállapítására, hogy javult-e a folyamat hatékonysága vagy sem. Figyelhető például a tesztelésre szánt munka és idő: a tesztelési folyamat tényleges javulásának csökkentenie kell a befektetett munkát, a tesztelési időt, vagy egyszerre mindkettőt. A folyamatmérések önmagukban azonban nem alkalmazhatók annak meghatározására, hogy javult-e a termék minősége. A termékmetrikákat (27. fejezet) össze kell gyűjteni, és össze kell kapcsolni a folyamattevékenységekkel.

A folyamatmetrikák három osztályba sorolhatók:

1. *Egy adott folyamat befejezéséig eltelt idő.* Ez lehet a folyamatra szánt teljes idő, a naptári idő, az egyes mérnökök által a folyamatra fordított idő stb.
2. *Egy adott folyamathoz szükséges erőforrások.* Erőforrásnak számíthat a teljes munka embernapokban, az utazások költségei, a számítógépes erőforrások stb.
3. *Egy adott esemény bekövetkezéseinek száma.* Megfigyelhető esemény lehet a kód vizsgálata során felfedezett hibák száma, a követelmények változtatásának száma, a követelményekben történt változtatások hatására megváltoztatott kódsorok száma stb.

Az első két osztályba tartozó mérések segíthetnek annak felderítésében, hogy a folyamaton végzett változtatások javították-e a folyamat hatékonyságát. Ha feltesszük, hogy a szoftverfejlesztési folyamatban bizonyos pontok (például a követelmények elfogadása, az architekturális tervezés befejezése, a tesztadatok generálásának befejezése stb.) rögzítve vannak, lehetséges mérni az egyik pontból a másikba való eljutáshoz szükséges időt és munkát. A mért értékek segítségével megnevezhetők azok a területek, ahol a folyamatot fejleszteni lehet, a változtatások bevezetése után pedig a rendszerjellemzők méréséből kiderülhet, hogy valóban hasznosak voltak-e a folyamaton végrehajtott változtatások.

Az eseményszám mérése közvetlenebb kihatással van a szoftverek minőségére. Ha például egy program ellenőrzési folyamatának megváltoztatása révén több hibát fedeznek fel a programban, akkor ez valószínűleg a jobb termékminőségben is tükröződni fog, de ezt a későbbi termékméréseknek is meg kell erősítenie.

A folyamatmérés alapvető problémája az, hogy tudni kell, mit is kell mérni. Basili és Rombach (Basili és Rombach, 1988) javasolták azt az általuk GQM-nek (Goal-Question-Metric – cél-kérdés-metrika) nevezett paradigmát, amely segít annak eldöntésében, hogy melyik méréseket válasszák, és ezeket hogyan kellene alkalmazni. A szemlélet arra támaszkodik, hogy azonosítják a következőket:

1. *Célok.* Cél az, amit a szervezet próbál megvalósítani (például megnövekedett programozói termelékenység, rövidebb termékfejlesztési idő, a termék megbízhatóságának növekedése stb.).
2. *Kérdések.* A kérdések a célok finomításai, ahol azonosítják a célokkal kapcsolatos meghatározott bizonytalansági területeket. Egy célhoz rendszerint számos megválaszolendő kérdés tartozik. A fent felsorolt célokhoz például a következő kérdések kapcsolódhatnak:

- Hogyan növelhető a hibátlan sorok száma?
 - Hogyan csökkenthető a termékkövetelmények véglegesítéséhez szükséges idő?
 - Hogyan lehet hatékonyabb megbízhatósági felméréseket végezni?
3. *Metrikák.* Ezek azok az összegyűjtendő mérések, amelyek segítenek a kérdések megválaszolásában, és amelyek alátámasztják, hogy a folyamat továbbfejlesztésével elérték-e a kitűzött célokat vagy sem. A fenti példákhoz olyan mérések készíthetők, amelyek mérik az egyéni programozók kódsorban megadott termelékenységet és gyakorlottságuk szintjét, a vásárló és a szállító közötti formális kommunikáció számát a követelmények minden egyes megváltoztatásakor vagy a termékhiba előidézéséhez szükséges tesztek számát.

A folyamat továbbfejlesztésére alkalmazott fenti megközelítés előnye, hogy elkülöníti a szervezethez kapcsolódó elemeket (a célokat) a speciálisan a folyamathoz kapcsolódó elemektől (a kérdésektől). Középpontjában az adatkapcsolat áll, és azt ajánlja, hogy az összegyűjtött adatokat a megválaszolandó kérdés függvényében különbözőképpen elemezzék. A módszer hosszú távú, mérésalapú folyamat-továbbfejlesztési programokban való alkalmazását Basili és Green (Basili és Green, 1993) írta le.

A GQM-szemléletet a szoftverfolyamat továbbfejlesztésének ami-modelljében (Pulford és mások, 1996) fejlesztették ki és egyesítették a SEI képességfejlettségi modellel (Paulk és mások, 1995). Az ami-módszer kifejlesztői a folyamat továbbfejlesztésére olyan szakaszos szemléletet javasoltak, amelyben akkor vezetnek be a mérést, amikor a szervezet valamilyen rendszert épített be a folyamataiba. A GQM irányt mutat, és gyakorlati tanácsokkal lát el a mérésalapú folyamat-továbbfejlesztés implementálásához.

28.4. A FOLYAMAT ELEMZÉSE ÉS MODELLEZÉSE

A folyamat elemzéséhez és modellezéséhez a meglévő folyamatokat kell tanulmányozni, és e folyamatok egy olyan absztrakt modelljét kell kifejleszteni, amely a folyamatok leglényegesebb jellemzőit gyűjti össze. Ezek a modellek segítenek a folyamat megértésében és ezen ismeretek továbbításában mások felé. A könyvben végig folyamatmodellek részleteit használtam fel az egyes tevékenységek (például követelmények tervezése, szoftvertervezés) tárgyalásánál. A 28.2. ábrán azt mutattam be, hogy a folyamat elemzése a mérés után jön. Ez egy leegyszerűsítés, mert a valóságban ezek a tevékenységek egybefonódnak. Az elemzéshez szükséges tudni, hogy mit mérünk, és a mérés végrehajtása közben mélyebb ismereteket szerzünk az éppen mért folyamatról.

A folyamatelemzés a meglévő folyamatokat vizsgálja a folyamat különböző részei közötti kapcsolat megértésének céljából. A folyamatelemzés első szakaszában elkerülhetetlenül a minőségi szempontokon van a hangsúly, és az elemzők egyszerűen a modell fő sajátosságait próbálják felfedni, míg a későbbi szakaszok-

ban már nagyobb mértékben alkalmaznak mennyiségi elemzést. A folyamatot automatikusan vagy manuálisan gyűjtött különféle metrikák alkalmazásával részletesebben is elemzik. Az elemzés után a folyamat leírható és dokumentálható egy folyamatmodellel (Huff, 1996).

A folyamatelemzés kiindulópontja lehet egy „formális” folyamatmodell. Sok szervezetnek olyan formális modellje van, amelyet a szoftver vásárlói írtak elő számára. Ez a szabvány határozza meg a kritikus tevékenységeket és azokat a résztermékeket, amelyeket az életciklusban elő kell állítani.

A formális modellek hasznos kiindulási pontként szolgálnak a folyamatelemzéshez, de a modellek ritkán elég részletesek, és nem tükrözik a szoftverfejlesztés valódi tevékenységeit. A formális folyamatmodellek meglehetősen absztraktak, és csak a fő folyamattevékenységeket vagy eredményeket határozzák meg, így általában a modellt alaposabban tanulmányozni kell, hogy megfeleljék a valóban lejátszódó folyamatokat. Ráadásul a ténylegesen alkalmazott folyamat gyakran jelentősen eltér a formális modelltől, jöllehet az előírt résztermékek általában elkészülnek.

A következő két technika a folyamatelemzési technikák közé tartozik:

1. *Kérdőívek és interjúk.* A tényleges történésekről a projekten dolgozó mérnököket kérdezik meg. A formális kérdőívre adott válaszokat a folyamatba bevont emberekkel készített személyes interjúk során finomítják.
2. *Etnográfiai tanulmányok.* Ahogyan azt a 7. fejezetben is mondtam, az etnográfiai tanulmányokat fel lehet használni a szoftverfejlesztés mint emberi tevékenység természetének megértésére. Ilyen elemzésekkel olyan részletek és bizonyult dolgok mutathatók ki, amelyek más technikákkal nem derülnének ki.

Mindkét technikának vannak előnyei és hátrányai. A kérdőívalapú elemzés elég gyorsan elvégezhető, ha egyszer már megállapították a megfelelő kérdéseket, de ha nem jól fogalmazták meg azokat, vagy nem a témára vonatkoznak, akkor a folyamat hiányos vagy téves modelljét eredményezik. Ezenkívül ez a technika olyan, mintha az értékelés egy formája lenne. A megkérdezett mérnökök olyan válaszokat is adhatnak, amelyek nem az igazságot tárják fel a folyamatról, hanem azt, amit szerintük a kérdező hallani szeretne.

Az interjú kimenete nyitottabb, mint a kérdőív. Ez egy előre elkészített kérdéssorozattal indul, de igazítható a különböző emberektől kapott válaszokhoz. Ha lehetőséget adunk a résztvevőnek egy téma részletesebb kifejtésére, akkor hallhatunk a folyamat problémáiról, arról, hogy a gyakorlatban hogyan változik a folyamat stb.

Az etnográfiai elemzésnél nagyobb az esély a valódi folyamat feltárására, viszont ez hosszadalmas és költséges tevékenység, amely több hónapig is eltarthat. Az etnográfiai elemzés a folyamat külső megfigyeléseire támaszkodik. A teljes elemzésnek a projekt kezdeti szakaszától egészen a termék átadásáig és karbantartásáig kell tartania, ami több évig tartó projekteknél nem mindig hajtható végre. Éppen ezért az etnográfiai elemzés akkor a leghasznosabb, ha a folyamat részeit mélységükben kell megérteni. A folyamat részleteinek megismeréséhez kisebb léptékű tanulmányozást válasszunk.

A folyamatelemzés eredménye a folyamatmodell, amely részletesen vagy kevésbé részletesen is kifejezhető. A könyvben használt folyamatmodellek absztrakt, egyszerűsített modellek, amelyek az adott folyamatok általános nézetét nyújtják. Az ilyen absztrakt szintű folyamatok sok szervezetnél megegyeznek, de ezeket az általános modelleket a fejlesztett szoftver típusától vagy a szervezeti környezettől függően különbözőképpen valósítják meg, ezért a részletesebb folyamatmodellek nem adhatók át a szervezetek között.

Az absztrakt vagy általános folyamatmodellek hasznos alapul szolgálnak a folyamatok tárgyalásához, de nem nyújtanak elegendő információt a folyamatelemzéshez és a folyamat továbbfejlesztéséhez, mert ehhez szükségesek a tevékenységekről, a résztermékekről, az emberekről, a kommunikációról, az ütemezésről és a szervezet más, a szoftverfejlesztési folyamatot befolyásoló folyamatairól szóló információk is. A részletes folyamatmodell elemeit a 28.6. ábra mutatja be.

Folyamatmodell-elem	Leírás
Tevékenység (jelölés: lekerekített sarkú téglalap, árnyék nélkül)	A tevékenységnek (például tesztadatok előkészítése egy modul teszteléséhez, egy függvény vagy modul kódolása, egy dokumentum korrigálása stb.) egyértelműen meghatározott feladata, belépési és kilépési feltétele van. A tevékenység általában atomi, vagyis egy személy vagy csoport felelősségébe tartozik, nem osztható fel altevékenységekre.
Folyamat (jelölés: lekerekített sarkú téglalap, árnyékkal)	A folyamat olyan tevékenységek együttese, amelyek valamilyen módon összetartoznak, és amelyeknek általános megegyezés szerinti feladata van a szervezeten belül (például a követelmények elemzése, architektúrális tervezés, teszttervezés).
Résztermék (jelölés: téglalap, árnyékkal)	A résztermék egy olyan tevékenység kézzelfogható kimenete, amelyet a projekttervben előre jeleztek.
Feltétel (jelölés: paralelogramma)	A feltétel lehet előfeltétel, amelynek a folyamat vagy tevékenység előtt, vagy utófeltétel, amelynek a folyamat vagy tevékenység befejezése után kell érvényesnek lennie.
Szerepkör (jelölés: kör, árnyékkal)	A szerepkör a felelősségek egy behatárolt területe (például konfigurációs vezető, tesztmérnök, szoftvertervező stb.). Egy személy több különböző szerepkörrel is rendelkezhet, és egy szerepkört több különböző ember is birtokolhat.
Kivétel (jelölés: nem szerepel az alábbi példákban, de duplakeretes téglalappal jelölhető)	A kivétel azt írja le, hogyan kell módosítani a folyamatot, ha valamilyen előre látható vagy előre nem látható esemény következik be. A kivételeket gyakran nem definiálják, és kezelésüket a projektvezető vagy a mérnökök leleményességére bízák.
Kommunikáció (jelölés: nyíl)	Emberek, vagy emberek és a támogató számítógépes rendszer közötti információcsere. A kommunikáció lehet formális vagy informális: formális kommunikáció az, amikor a projektvezető jóváhagyja az átadandó terméket; informális kommunikáció pedig az, amikor elektronikus levelet váltanak egy dokumentum nem egyértelmű részeinek tisztázására.

28.6. ábra. A folyamatmodell elemei

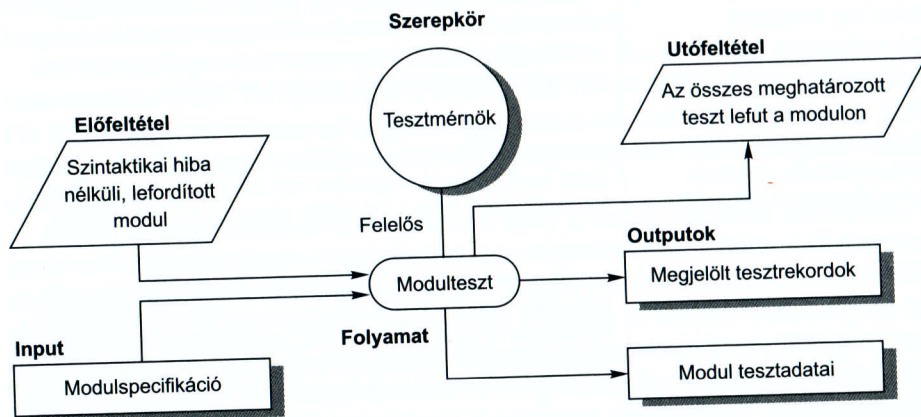
A tevékenységek, a résztermékek és a kommunikáció időzítését és a köztük lévő függőségeket szintén fel kell tüntetni a folyamatmodellben. A tevékenységek néha párhuzamosan, néha szekvenciálisan hajthatók végre, akár egymást átfordva, így ugyanaz a mérnök több tevékenységben is részt vehet. A résztermékek függhetnek más résztermékektől vagy a projekten dolgozó mérnökök közötti kommunikációtól.

A könyvben szereplő folyamatmodellek példáiban a tevékenységek hozzávetőleges sorrendje balról jobbra haladva követhető, a párhuzamosan végezhető tevékenységeket a lehetőségekhez mérten függőlegesen ábrázoltam.

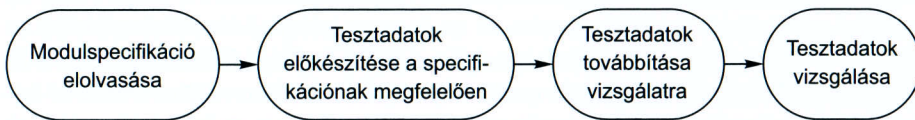
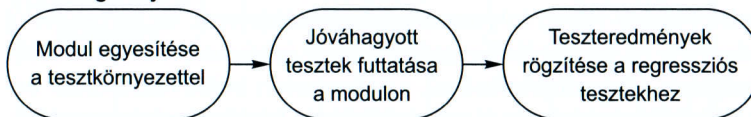
A részletes folyamatmodellek rendkívül összetettek. Példaként tekintsük a 28.7. és a 28.8. ábrát. Ezek egy szigorúan ellenőrzött konfigurációkezelési folyamatot (29. fejezet) használó nagy rendszer egyetlen moduljának tesztelési folyamatát írják le. A tesztelés alatt álló szoftver és a tesztadatok konfiguráció-ellenőrzés alatt állnak. A tesztelési folyamatért, a folyamat bemeneteiért és kimeneteiért, valamint az elő- és utófeltételekért felelős szerepkör a 28.7. ábrán látható.

A „modulteszt” folyamat számos különálló tevékenységre bontását a 28.8. ábra mutatja, de csak a modultesztelés viszonylag egyszerű tevékenységeiben szereplő tevékenységeket ábrázolja. Ez a négy tevékenységfolyam a tesztadatok előkészítése, tesztkörnyezet írása a modulhoz, a teszt futtatása és a tesztről szóló jelentés elkészítése. Az előkészítési folyamat tevékenységei rendszerint más folyamatok alatt is elvégezhetők, az előkészítési tevékenységeknek pedig nyilvánvalóan meg kell előzniük a végrehajtási és a jelentéskészítési tevékenységeket.

Az ábrán nem tüntettem fel a folyamat elő- és utófeltételeit, valamint az információkat a folyamat bemeneteiről és kimeneteiről, mert ezek csak bonyolítanak a modellt és nehezítenék a megértését. Ahelyett, hogy egyetlen modellbe próbálnánk besűríteni az összes információt, inkább több különböző, más-más absztrakciós szintű modellt ajánlott készíteni, amelyek közös elemeik (például tevékenységek, résztermékek) révén kapcsolatban állnak egymással. A modellek közül néhánynak elsősorban a folyamattevékenységekre, másoknak a folyamat végrehajtását irányító vezérlőinformációkra kellene vonatkoznia.



28.7. ábra. A modulteszt folyamata

Tesztadatok előkészítése**Modulteszt előkészítése****Teszt végrehajtása****Tesztjelentés****28.8. ábra.** A modulteszt tevékenységei**28.4.1. A folyamat kivételes helyzetei**

A szoftverfolyamatok nagyon bonyolult entitások. Bár egy szervezetnek lehet definiált folyamatmodellje, ez csakis arra az ideális esetre érvényes, amikor a fejlesztőcsapat nem ütközik váratlan problémákba. A valóságban a váratlan problémák mindennaposak a projektvezetők életében, az „ideális” folyamatmodellt pedig dinamikus módon módosítani kell, amint megtalálták a megoldásokat ezekre a problémákra. A projektvezetők által kezelendő kivételes helyzetek lehetnek például a következők:

1. számos kulcsembert betegedik meg egyidejűleg, éppen egy kritikus projekt-felülvizsgálat előtt;
2. komolyan megsértik a biztonságot, így több óráig nem működik a külső kommunikáció;
3. a szervezet átalakul, azaz a vezetőknek idejük nagy részében nem a projekt vezetésével, hanem szervezeti ügyekkel kell foglalkozniuk;
4. nem várt kérés érkezik új projektjavaslatok tételére. A javaslat kidolgozásához a projekttől kell elvonni a munkát.

A kivételes helyzetek hatására általában valamilyen módon megváltoznak a projekt erőforrásai, a költségvetése és az ütemezése. Nehéz minden kivételes helyzetet előrejelezni, és formális folyamatmodellbe foglalni őket, ezért a vezetők sok kivételes helyzetet a „szabványos” folyamat dinamikus változtatásával kezelnek, hogy az a váratlan helyzetekben is megállja a helyét. A folyamatmodellek természetesen nem teljeseek, és a folyamatmenedzser a felelős a kivételek kezeléséért és a szükséges folyamatadaptációkért.

28.5. A FOLYAMAT MEGVÁLTOZTATÁSA

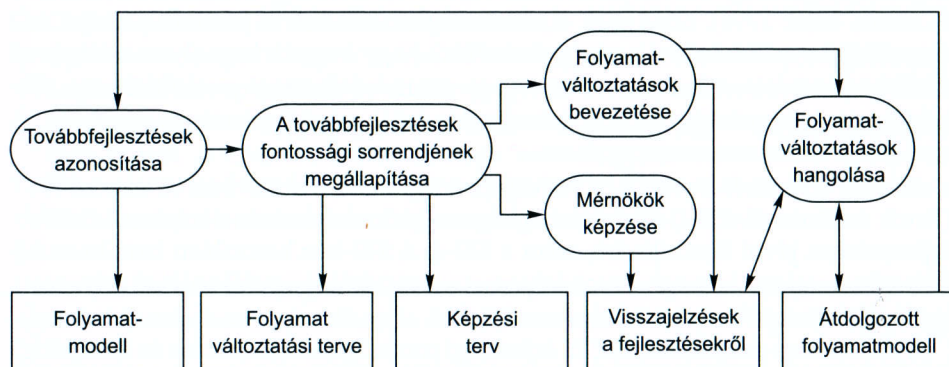
A folyamat megváltoztatása létező folyamat módosítását jelenti. Ez történhet új gyakorlatok, módszerek és eszközök bevezetésével, a folyamaton belül a tevékenységek sorrendjének megváltoztatásával, a folyamaton belül résztermék bevezetésével vagy eltávolításával, vagy új szerepkörök és felelősségek bevezetésével. A folyamat-továbbfejlesztés célja lehet például „25 százalékkal csökkenteni az integrációs tesztelés során feltárt rendellenességek számát”. Ez a cél vezérelheti a folyamatváltoztatást, és a változtatások bevezetése után használható a mérés fejlesztésére is.

A folyamatváltoztatási folyamat öt kulcslépésből áll (28.9. ábra):

1. *A továbbfejlesztés azonosítása.* Ebben a lépésben a folyamatelemzés eredményeinek felhasználásával megállapíthatók, melyek azok a minőségi, ütemezési vagy költségbeli akadályok, amelyeknél a folyamatényezők hátrányosan befolyásolhatják a termék minőségét. A folyamat továbbfejlesztésénél arra kell összpontosítani, hogy elhárítsák ezeket az akadályokat új eljárások, módszerek és eszközök ajánlásával.
2. *A továbbfejlesztés fontossági sorrendjének megállapítása.* Ez a lépés a lehetséges változtatások felmérésére és azok fontossági sorrendjének megállapítására vonatkozik. Ha több lehetséges változtatást azonosítottunk, akkor azok egyidejű bevezetése lehetetlen; el kell döntenie, melyek a legfontosabbak. Ez a döntés alapulhat a specifikus folyamatterületek továbbfejlesztésének fontosságán, a változtatások bevezetésének költségein, a változtatás szervezetre gyakorolt hatásán, vagy más tényezőkön.
3. *A folyamat változtatásainak bevezetése.* A folyamat változtatásainak bevezetése az új eljárások, módszerek és eszközök alkalmazását és ezeknek a folyamat más tevékenységeibe való integrálását jelenti. Fontos, hogy elegendő idő álljon rendelkezésre a változtatások bevezetéséhez, és hogy biztosítsák a változtatások más folyamattevékenységekkel, a szervezet eljárásaival és szabványaival való kompatibilitását.
4. *A folyamatváltoztatások betanítása.* Betanítás nélkül nem lehet kiaknázni a folyamatváltoztatások valamennyi előnyét. Megeshet, hogy a vezetők és a fejlesztési folyamatért felelős mérnökök nem vesznek tudomást erről a lépésről. Nagyon is gyakran fordul elő, hogy a folyamat változtatásait a kellő betanítás

nélkül kezdik alkalmazni, ezért annak hatásai inkább rontják, mintsem javítják a termék minőségét.

5. *A változtatások hangolása.* A folyamat javasolt változtatásai sosem lesznek közvetlenül a bevezetésük után maximálisan hatékonyak. Szükség van olyan hangolási fázisra, amely során kiderülnek a kisebb problémák, valamint módosításokat javasolnak és vezetnek be. A hangolási fázis több hónapig is eltart, amíg a fejlesztőmérnökök elégedettek nem lesznek az új folyamattal.



28.9. ábra. A folyamatmegváltoztató folyamat

Amint bevezettek egy változtatást, a továbbfejlesztési folyamatot újra meg lehet ismételni, további elemzés segítségével azonosíthatók a problémák, javítások tehetők stb. Nem praktikus, ha egyszerre túl sok változtatást vezetnek be: azonkívül, hogy ez a betanítás során problémákhoz vezethet, még az egyes változtatásoknak a folyamatra gyakorolt hatását is lehetetlen felmérni.

Egy vezetőnek a folyamatváltoztatások bevezetésénél tekintettel kell lennie a csapatába tartozó emberek érzéseivel. Az üzleti folyamatok újratervezése (Hammer, 1990) – az 1990-es években divatos, radikális folyamatváltoztatásokat okozó módszere – azért volt sikertelen, mert nem vette figyelembe az emberek napi gyakorlatát. Az emberek úgy érezték, hogy a szakértelmüket nem veszik figyelembe, és a munkájukat úgy változtatják meg, hogy azt nem beszélik meg velük. Aktívan ellenálltak a változtatásoknak, és elérték, hogy azok ne működjenek.

Nem kérdés, hogy egyes emberek fenyegetőnek érzik a változtatásokat, úgy érzik, elveszíthetik a munkájukat, vagy nem fognak megfelelni az új elvárásoknak. A csapatot be kell vonni a folyamatváltoztatásokba, meg kell érteni a kétélyeiket, és be kell vonni őket az új folyamat tervezésébe. A folyamatváltoztatások részesévé kell tenni őket, és még inkább, hogy akarják azt megvalósítani a saját munkájukon keresztül.

28.6. A CMMI FOLYAMAT-TOVÁBBFEJLESZTŐ KERETRENDSZER

A SEI (Software Engineering Institute) létrehozásának célja az Egyesült Államok szoftveripari lehetőségeinek javítása volt. A SEI az 1980-as évek közepén kezdett bele annak tanulmányozásába, hogyan lehet felbecsülni a szállítók lehetőségeit. E képességfelmérési munka eredményeként született meg a SEI Szoftverképeség Fejlettségi Modell (Software Capability Maturity Model, CMM) (Paulk és mások, 1993; 1995), amelynek általánosságban rendkívül jelentős szerepe volt a szoftvertervező közösség meggyőzésében, hogy vegyék komolyan a folyamat továbbfejlesztését. A Software CMM-t egy sor más fejlettségi modell követte, többek között a People Capability Maturity Model (P-CMM) (Curtis és mások, 2001), amelyet a 25. fejezetben tárgyaltam.

Más szervezetek hasonló folyamatfejlettségi modelleket fejlesztettek. A SPICE (Paulk és Konrad, 1994) közelítése a képességfelméréshez és a folyamat-továbbfejlesztéshez jóval flexibilisebb, mint a SEI-é. A SEI-hez hasonlóan tartalmaz fejlettségi szinteket, de meghatároz folyamatokat (például ügyfél-szállító folyamat), amelyek keresztül metszik ezeket a szinteket.

A Bootstrap-projekt célja a SEI fejlettségi modelljének kibővítése és átalakítása volt, hogy szervezetek szélesebb körében legyen alkalmazható. A Bootstrap-modell (Haase és mások, 1994; Kuvaja és mások, 1994) a SEI-modell fejlettségi szintjeit használja, de a következőket is megcélozza:

1. iránymutatás a teljes vállalatra kiterjedő, a folyamat továbbfejlesztését támogató minőségi rendszerhez;
2. a szervezet, a módszertan és a technológia közötti fontos különbség hangsúlyozása;
3. egy átvethető, alapvető folyamatmodell (amely a European Space Agency által használt modellen alapszik).

A kifejlesztett modellek sokasága (beleértve a sajátját is) miatt a SEI elhatározta egy integrált fejlettségi modell (CMMI) kidolgozását. Ez egyesíti a szoftver- és rendszerképeség-fejlettségi modelleket és integrálja azokat más fejlesztési modellekkel. Két fajtája van: a lépcsős és a folytonos, és megszünteti a szoftverképeség-fejlettségi modell gyengeségeit.

A CMMI-modell (Ahern és mások, 2001) egy folyamat továbbfejlesztési keretrendszer, amely a vállalat egészében használható. A lépcsős verziója kompatibilis a szoftverképeség-fejlettségi modellel és lehetővé teszi a szervezeti rendszerfejlesztési és menedzselési folyamatok felmérését és osztályozását 1-től 5-ig terjedő fejlettségi szinten. A folytonos modell finomabb osztályozást tesz lehetővé 24 területen, 1-től 6-ig terjedő skálán (28.10. ábra).

A modell nagyon bonyolult (a leírása több mint ezer oldal), így én itt radikálisan leegyszerűsíttem a bemutatását:

1. *Folyamatterületek.* A CMMI 24 folyamatterületet sorol fel, amelyek a szoftver-folyamat fejlettségére és továbbfejlesztésére vonatkoznak. A folytonos CMMI-modellben ezeket négy csoportba sorolják. Ezeket a csoportokat mutatja be a 28.10. ábra.
2. *Célok.* A célok egy szervezet által elérendő kívánt állapot absztrakt leírásai. A CMMI specifikus céljai hozzá vannak rendelve a folyamatterületekhez, és meghatározzák a területek kívánt állapotait. Definiál általános célokat is, amelyek a legjobb gyakorlatok bevezetéséhez kapcsolódnak. A 28.11. ábra a CMMI specifikus és általános céljait mutatja.
3. *Gyakorlatok.* A gyakorlatok a CMMI-ben a célok elérésének útját adják meg. Legfeljebb hét specifikus és általános gyakorlat kapcsolható minden célhoz minden folyamatterületen belül. Példákat az ajánlott gyakorlatokra a 28.12. ábra mutat. A CMMI felismerte, hogy nem a cél, hanem a hozzá vezető út a fontos. A szervezetek bármilyen megfelelő gyakorlatot alkalmazhatnak a CMMI-célok elérésére – nem kell megfogadni a CMMI-ajánlásokat.

Az általános célok és gyakorlatok nem technikaiak, hanem a jó gyakorlat vállalati bevezetéséhez kapcsolhatók. Mindez a szervezet fejlettségétől függ. Egy

Kategória	Folyamatterületek
Folyamatkezelés	Szervezeti folyamat meghatározása Szervezeti folyamat kijelölése Szervezeti képzés Szervezeti folyamat hatékonysága Szervezeti innováció és bevezetés
Projektmenedzsment	Projekttervezés Projekt követése és irányítása Beszállítói megállapodások kezelése Integrált projektmenedzsment Kockázatkezelés Integrált csapatmunka Kvantitatív projektmenedzsment
Fejlesztés	Követelménykezelés Követelményfejlesztés Technikai megoldások Verifikáció Validáció
Támogatás	Konfigurációkezelés Folyamat és termék minőségkezelés Mérés és elemzés Döntéselemzés és döntés Integrációs szervezeti környezet Okelemzés és döntés

28.10. ábra. A CMMI-folyamatterületek

Cél	Folyamatterületek
Korrektív tevékenységek a befejezésre, ha a projekt hatékonysága vagy eredményei lényegesen eltérnek a tervezettől.	Specifikus cél a projektfigyelésben és -irányításban.
A projekt aktuális teljesítményének és előrehaladásának figyelése a projekttervhez képest.	Specifikus cél a projektfigyelésben és -irányításban.
A követelmények elemzése és validálása és a fejlesztendő funkcionálisok meghatározása.	Specifikus cél a követelmény-fejlesztésben.
A hiányosságok és egyéb problémák okainak szisztematikus felderítése.	Specifikus cél az okelemzés és döntés területén.
A folyamat véglegesített folyamatként való bevezetése.	Általános cél.

28.11. ábra. Példák a CMMI céljaira

Gyakorlat	Hozzárendelt célok
Az elemzésből származó követelményekről bizonyítani kell, hogy azok szükségesek és hasznosak. A követelményeket validálni kell, hogy biztosítsuk, a végtermék beilleszthető a megfelelő, többféle technikát alkalmazó felhasználói környezetbe.	A követelmények elemzése és validálása, és a fejlesztendő funkcionálisok meghatározása.
A hiányosságok és egyéb problémák kiválasztása elemzésre. A kiválasztott hiányosságok és egyéb problémák okainak elemzése, és javaslat a kiküszöbölésükre.	A hiányosságok és egyéb problémák okainak szisztematikus felderítése.
Szervezeti politika kialakítása és kezelése a követelmények fejlesztési folyamatának tervezéséhez és végrehajtásához. Felelősség és jogosultság adása a folyamat végrehajtásához, a munkatermékek kifejlesztése és a fejlesztési folyamat szolgáltatásainak biztosítása.	A folyamat véglegesített folyamatként való bevezetése.

28.12. ábra. A CMMI gyakorlatai és a hozzájuk rendelt célok

fiatal, a fejlettség kezdeti lépcsőjén álló szervezet számára a bevezetés már meghonosodott tervek és folyamatok alkalmazását jelenti. Ugyanakkor egy fejlettebb, kialakult folyamatokkal bíró szervezet már vezérelheti a folyamatait statisztikai vagy más mennyiségi technikákkal.

Egy CMMI-felmérés a szervezet folyamatainak megvizsgálását és minden folyamatterület egy hatfokozatú skálán való elhelyezését jelenti. A hatfokozatú skála jelentése a következő:

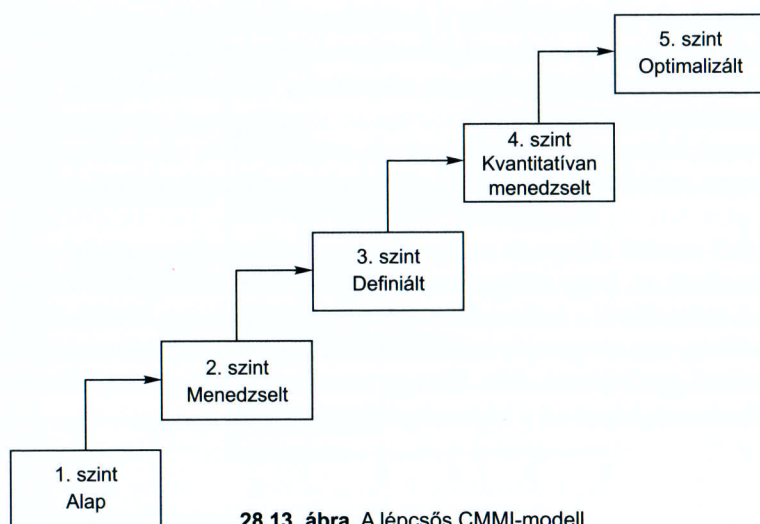
1. *Nem végrehajtható.* Egy vagy több folyamatterülethez rendelt specifikus cél nem létezik.
2. *Végrehajtható.* A folyamatterülethez rendelt specifikus célok megvannak, és minden csapattaggal explicit módon közölték a folyamat szerepét minden elvégzendő munkában.

3. *Menedzsel*. Ezen a szinten a folyamatterülethez rendelt célok összehangoltak, és vannak olyan szervezeti politikák, amelyek megadják, mikor kell a folyamatokat használni.
4. *Definiált*. Ez a szint a szervezeti szabványokra és a folyamatok alkalmazására fókuszál. A szervezet minden projektje menedzsel folyamat, ami szervezeti folyamatok definiált halmazából áll. A folyamatgyűjtemények és folyamatmérések léteznek a folyamat jövőbeni továbbfejlesztéséhez.
5. *Kvantitatívan menedzsel*. Ezen a szinten a szervezet statisztikai és egyéb kvantitatív módszereket alkalmaz az alfolyamatok vezérlésére. Ez azt jelenti, hogy a folyamatgyűjtemények és folyamatmérések alkalmazása kötelező a projektmenedzsmentben.
6. *Optimalizált*. Ez a legmagasabb szint, a szervezetnél kötelező a folyamat és a termék mérése a folyamat továbbfejlesztéséhez. Elemezni kell a trendeket, és a folyamatot hozzá kell igazítani a változó üzleti szükségletekhez.

Természetesen a fejlettségi szintek fenti leírása egyszerűsített, de gondoljunk úgy a szintekre, mint egy fejlődő, a folyamat és a szoftver mérése által vezérelt explicit folyamatra, amely a legalacsonyabb szinttől a folyamatváltoztatások és -továbbfejlesztések szabványosításáig terjed.

28.6.1. A lépcsős CMMI-modell

A lépcsős CMMI modell a szoftverképesség-fejlettségi modellhez hasonló, a szervezet folyamatképességét öt szintbe sorolja. Felsorolja az egyes szinteken elérendő célokat. A folyamat továbbfejlesztése minden szinten a gyakorlatok implementálását jelenti, az alacsonyabb szinttől a magasabb felé mozogva a modellben. A lépcsős CMMI öt szintje a 28.13. ábrán látható.



28.13. ábra. A lépcsős CMMI-modell

Minden fejlettségi szinthez tartozik egy folyamatterület halmaza, illetve általános célok. Például a második (menedzselt) szinthez rendelt folyamatterületek a következők:

1. *Követelménykezelés.* Kezeli a projekttermék és a projektkomponensek követelményeit, és azonosítja a követelmények, a projektterv és a munkatermék közötti ellentmondásokat.
2. *Projekt tervezése.* Előállítja és karbantartja a tervet, ami definiálja a projekt tevékenységeit.
3. *Projekt figyelése és vezérlése.* Segít megérteni a projekt előrehaladását úgy, hogy a szükséges beavatkozásokat meg lehessen tenni, ha a projekt teljesítménye lényegesen eltér a tervtől.
4. *Beszállítói támogatások kezelése.* Kezeli a külső beszállítók termékeinek és szolgáltatásainak beszerzését, ha létezik formális megállapodás.
5. *Mérés és elemzés.* Kifejleszt és működtet egy mérési kapacitást, amely a szükséges vezetői információkat szolgáltatja.
6. *Folyamat- és termék-minőségbiztosítás.* Biztosítja a személyzet és a vezetés számára a folyamatok és a munkatermékek objektív megítélését.
7. *Konfigurációkezelés.* Létrehozza és fenntartja a munkatermék integritását a konfiguráció azonosításának, ellenőrzésének, státusnyilvántartásának és követésének segítségével.

Ha ezen specifikus gyakorlatok szerint a szervezet a CMMI-modell második szintjén dolgozik, akkor el tudja érni általános célként, hogy az összes folyamat menedzselt legyen. Például ahhoz, hogy a projekttervezéshez kapcsolt gyakorlat egy menedzselt projekttervezési folyamathoz vezessen, a következők kelljenek:

- Vállalati politika kialakítása és működtetése a projekttervezési folyamat megtervezéséhez és bevezetéséhez.
- Megfelelő erőforrások hozzárendelése a projektmenedzselési folyamathoz, a termék kifejlesztése és a folyamat szolgáltatásainak támogatása.
- A projekttervezési folyamat figyelése és ellenőrzése a tervhez képest, és a szükséges beavatkozások végrehajtása.
- A projekttervezési folyamat tevékenységeinek, státusának és eredményeinek áttekintése magas szintű menedzselés és döntések meghozása mellett.

A lépcsős CMMI-modell előnye (a szoftverképeség-fejlettségi modellel való kompatibilitása mellett) az, hogy világos továbbfejlesztési lehetőséget ad a szervezetek kezébe. A másodikról a harmadik szintre léphetnek és így tovább. Hátránya, hogy lehetőség van magasabb szintű célok és gyakorlatok bevezetésére az alacsonyabb szintű gyakorlatok előtt. Ha egy szervezet ezt megteszi, akkor a fejlettség felmérése hamis képet ad a képességeiről.

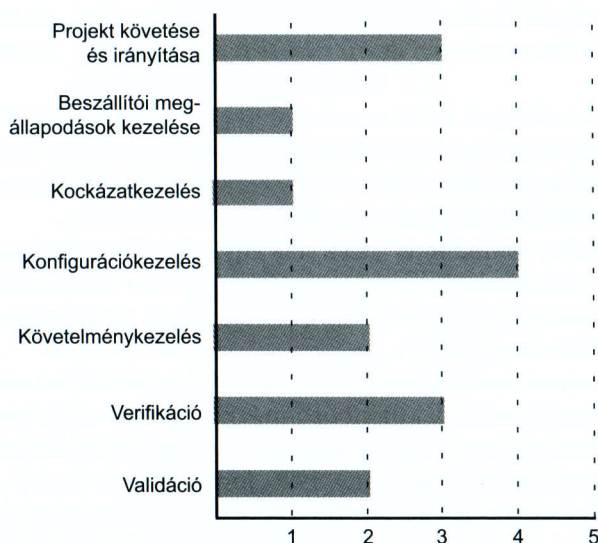
28.6.2. A folytonos CMMI-modell

A folytonos fejlettségi modell a szervezeteket nem diszkrét szintekkel jellemzi. Ez finomabb szemcsézettségű modell, amely egyedi gyakorlatokat vagy azok csoportját kezeli, és minden gyakorlat használatát figyelembe veszi. A fejlettség felmérésének eredménye nem egyetlen érték, hanem egy értékhalmoz, amely a szervezet minden folyamatának vagy folyamatcsoportjának fejlettségét mutatja.

A folytonos CMMI-modell besorol minden folyamatterületet (28.10. ábra), és mindegyiknek ad egy képességfelmérési szintet 1-től 6-ig (ahogy azt mindjárt láthatjuk).

Normális esetben a szervezetek különböző fejlettségi szinten dolgoznak a különböző folyamatterületeken. Következésképpen a folytonos CMMI-felmérés eredménye egy képességprofil, amely minden folyamatterületen a kapcsolódó képességet mutatja. A 28.14. ábra egy képességprofilrészlet mutat, a folyamatok különböző képességi szintjeivel. A szervezetek aktuálisan azt a képességprofilt fejleszthetik, amelyik azt a folyamatterületet tartalmazza, amelyet fejleszteni szeretnének.

A folytonos modell legalapvetőbb előnye az, hogy a szervezetek a saját szükségleteiknek és a követelményeknek megfelelő folyamatokat tudják kiragadni és továbbfejleszteni. Tapasztalataim szerint a különböző típusú szervezetek különböző követelményeket támasztanak a folyamat továbbfejlesztésével szemben. Például míg egy légi közlekedési szoftvereket gyártó vállalat a továbbfejlesztésben a rendszer-specifikációra, konfigurációkezelésre és validációra fókuszál, addig egy webfejlesztő vállalat számára az ügyfélkezelő folyamatok fontosak. A lépcsős modellben a vállalatoknak a különböző lépcsőkre kell koncentrálniuk. Ezzel szemben a folytonos CMMI több mérlegelést és hajlékonyságot tesz lehetővé.



28.14. ábra. A folyamat képességprofilja

Kulcsfogalmak

- A folyamat továbbfejlesztése magában foglalja a folyamatok elemzését, szabványosítását, mérését és megváltoztatását. A továbbfejlesztési folyamat hatékonnyá tételéhez fontos a folyamat betanítása.
- Egy folyamat besorolható az informális, menedzselte, tervszerű vagy tökéletesítő folyamatok közé. Ez az osztályozás a folyamatot támogató eszközök meghatározásánál használható fel.
- A folyamat-továbbfejlesztési ciklus magában foglalja a folyamatok mérését, elemzését, modellezését és megváltoztatását.
- A mérés a használt szoftverfolyamat specifikus kérdéseire adhat választ. Ezek a kérdések a szervezet továbbfejlesztési céljain alapulhatnak.
- A mérésnél használt folyamatmetrikáknak három típusa van: időmetrikák, erőforrás-felhasználási metrikák és eseménymetrikák.
- A folyamatmodell tartalmazza a tevékenységeket, az alfolyamatokat, a szerepköröket, a kivételeket, a kommunikációt, a résztermékeket és más folyamatok leírását.
- A méréseket az alkalmazott szoftverfolyamattal szemben felmerült sajátos kérdések megválaszolására ajánlott használni, amely kérdéseknek a szervezet továbbfejlesztési céljain kell alapulniuk.
- A CMMI folyamatfejlettségi modell egy integrált folyamat-továbbfejlesztési modell, amely támogatja mind a lépcsős, mind a folytonos folyamat-továbbfejlesztést.
- A folyamat-továbbfejlesztés a CMMI-modellben a jó szoftvertervezési gyakorlathoz kapcsolódó célok összegyűjtésén, és a célok eléréséhez használt leíráson, szabványosításon, ellenőrzésen alapszik. A CMMI-modell alkalmazható gyakorlatokat ajánl, amelyek használata azonban nem kötelező.

További irodalom

- CMMI Distilled.* A CMMI tömör összefoglalója. Könnyen olvasható azoknak, akik már értik a CMM fogalmát, egy általános bevezető a folyamat-továbbfejlesztésbe. (Ahern, D. H. és mások, 2001, Addison-Wesley.)
- „Can you trust software capability evaluations”. Ez a cikk egy szkeptikus áttekintése a képességiértékelés témakörének, és azt tárgyalja, hogy ez a kiértékelés nem adhat valós képet a szervezet fejlettségéről. (O’Connell, E. és Saiedian, H., *IEEE Computer*, 33(2), February 2000.)
- Trends in Software: Software Process Modelling and Technology.* A könyv kitűnő válogatást tartalmaz a szoftverfolyamatok különböző megközelítéseivel (folyamatmodellezés, folyamatmegtámogatás, a CMM alkalmazása) foglalkozó áttekintő tanulmányokból. (Fuggetta, A. és Wolf, A. (eds.), 1996, John Wiley & Sons.)

Feladatok

- 28.1. Javasoljon folyamatmodellezéseket a következő folyamatokhoz:
 - tűzrakás fából,
 - háromfogásos étel elkészítése (tetszés szerinti ételekkel),
 - rövid (50 soros) program írása.
- 28.2. Milyen körülmények között valószínűsíthető, hogy a fejlesztőcsapat határozza meg a termék minőségét? Mondjon példát olyan típusú szoftvertermékekre, amelyek nagymértékben függenek az egyéni tehetségtől és képességektől.
- 28.3. Magyarázza meg, hogy a 28.2. alfejezetben bevezetett tervszerű folyamat miért nem feltétlenül menedzselt folyamat.
- 28.4. Javasoljon három specializált szoftvereszközt, amelyek egy szervezetben kifejleszthetők a folyamat-továbbfejlesztés támogatására.
- 28.5. Tegyük fel, hogy egy szervezetben a folyamat-továbbfejlesztés célja a fejlesztés során előállított újrafelhasználható komponensek számának növelése. Javasoljon három, a célra vonatkozó kérdést a QQM-paradigmában.
- 28.6. Írjon le háromféle szoftvermetrikát, amely a folyamat-továbbfejlesztési tevékenység részeként összegyűjthető. Minden metrikatípusra adjon példát.
- 28.7. Tervezzon egy folyamatot a folyamatváltoztatási javaslatok felmérésére és fontossági sorrendbe állítására. Dokumentálja ezt a folyamatot egy folyamatmodellel, megadva a folyamat szerepköreit.
- 28.8. Adja meg két előnyét és hátrányát a folyamatfelmérés és -továbbfejlesztés olyan közelítéseinek, amelyek be vannak ágyazva olyan folyamat-továbbfejlesztési keretrendszerekbe, mint a CMMI.
- 28.9. Milyen körülmények között javasolná a CMMI lépcsős reprezentációjának használatát?
- 28.10. Mik a különbségek a CMMI általános és specifikus céljai között?
- 28.11. Mik az előnyei és hátrányai, ha célorientált fejlettségi modellt használunk gyakorlatorientált helyett?
- 28.12. Alapjában véve figyelmen kívül hagyják-e az emberi tényezőt azok a folyamat-továbbfejlesztési programok, amelyek az embereknek a folyamatoknál és a folyamatok változtatásánál végzett munkáját mérik? Milyen ellenállás léphet fel egy ilyen folyamat továbbfejlesztési programmal szemben?