

27 A MINŐSÉG KEZELÉSE

TÉMAKÖRÖK

A fejezet bevezetést nyújt a szoftverminőség kezelésébe és a szoftver mérésébe. A fejezet elolvasása után:

- megértjük a minőségkezelési folyamatot és a minőségbiztosítási, minőségtervezési és minőség-ellenőrzési folyamatok fő tevékenységeit;
- megértjük, mekkora jelentőségük van a szabványoknak a minőségkezelési folyamatban;
- megértjük a szoftvermetrika fogalmát, valamint a prediktor- és vezérlési metrikák közti különbséget;
- megértjük, hogyan segíthetnek a mérések a minőségi jellemzők és a szoftvermérés jelenlegi korlátozásainak felmérésében;
- megismerjük a szoftvermérés aktuális korlátait.

TARTALOM

- 27.1. A folyamat és a termék minősége
- 27.2. A minőségbiztosítás és a szabványok
- 27.3. A minőségtervezés
- 27.4. A minőség-ellenőrzés
- 27.5. A szoftver mérése és a metrikák

A szoftver minősége az elmúlt 15 év alatt szignifikánsan megnőtt. Ennek egyik oka az, hogy a vállalatok új technikákat és technológiákat vezettek be, olyanokat, mint például az objektumorientált fejlesztés és a hozzá tartozó CASE-támogatás. Ráadásul egyre inkább tudatosul a szoftver minőségkezelésének és a szoftvergyártásban alkalmazott minőségkezelési technikáknak a jelentősége.

A szoftverminőség azonban összetett fogalom, amely közvetlenül nem összehasonlítható az iparbeli minőséggel. Az iparban a minőség fogalma azt jelenti, hogy a kifejlesztett termék feleljen meg a specifikációjának (Crosby, 1979). Egy ideális világban ez a definíció alkalmazható is lenne minden termékre, viszont szoftverrendszerek esetében problémák adódnak:

1. A specifikációnak a vásárló terméktől elvárt jellemzőit kell megcélózni, de emellett a fejlesztő szervezetnek is lehetnek követelményei (például karbantarthatósági követelmények), amelyek nem szerepelnek a specifikációban.
2. Bizonyos minőségi jellemzőket (például karbantarthatóság) nem lehet egyértelműen definiálni.
3. Ahogyan azt a követelmények tervezéséről szóló 1. részben kifejtettem, nagyon nehéz teljes szoftverspecifikációt írni, ezért attól, hogy egy szoftver megfelel a specifikációjának, nem biztos, hogy felhasználói jó minőségűnek fogják tartani, mivel az nem felel meg az elvárásaiknak.

A problémákat a meglévő szoftverspecifikációkban kell felfedezni, és olyan minőségkezelő eljárásokat kell tervezni, amelyek nem pontos specifikációval rendelkeznek. Főleg az olyan szoftverjellemzőket, mint például karbantarthatóság, biztonság, hatékonyság, nem lehet explicit módon specifikálni. Ugyanakkor ezek nagyban befolyásolják a rendszer érzékelhető minőségét. Ezeket a jellemzőket a 27.3. alfejezetben tárgyalom.

Egyesek azt gondolják, hogy a minőség elérhető szabványok meghatározásával és olyan szervezeti minőségi eljárásokkal, amelyek ellenőrzik, hogy ezeket a szabványokat alkalmazza-e a szoftverfejlesztő csapat. Ők azzal érvelnek, hogy a szabványok a jó gyakorlatot tartalmazzák, és ezt a jó gyakorlatot követve kiváló minőségű termékek jönnek létre. Azt gondolom, hogy a minőségkezelés sokkal több, mint a szabványok és az azok betartatását biztosító bürokrácia.

A jó minőségi vezető olyan „minőségi kultúra” kialakítására törekszik, ahol mindenki, aki a termék fejlesztéséért felelős, kiáll a magas szintű termékminőség mellett. Serkenti a csapatokat, hogy vállaljanak felelősséget munkájuk minőségéért, és fejlesszenek ki új szemléletmódokat a minőség fejlesztésére. Noha a minőségkezelés alapját az eljárások és a szabványok adják, a gyakorlott minőségi vezetők elismerik, hogy a szoftverminőségnek vannak olyan megfoghatatlan oldalai (elegancia, olvashatóság stb.), amelyek nem szabványosíthatók. A jó minőségi vezető támogatja az ezek iránt érdeklődő embereket, és a csapat minden tagját ösztönzi a professzionális magatartásra.

A formalizált minőségkezelés különösen fontos a nagy és komplex rendszereket fejlesztő csapatok számára. A minőség dokumentációja annak a feljegyzése, hogy a projektben az alcsoportok mit csináltak. Segít az embereknek abban, hogy ellenőrizni lehessen, hogy fontos feladatok nem felejtődtek el, illetve hogy a csapat egy része nem él helytelen feltételezésekkel arról, hogy mások mit csinálnak. A minőség dokumentációja egyben az eszköze a kommunikációnak is egy rendszer élettartama alatt. Megállapítható belőle a csoportok felelőssége a rendszer evolúcióban, mivel nyomon követi a fejlesztőcsapat munkáját.

A minőségkezelés kisebb rendszerek esetén is fontos, de egy informálisabb közelítést lehet alkalmazni. Nem szükséges akkor a papírmunka, mivel egy kis fejlesztőcsapat informálisan is tud kommunikálni. A kisméretű rendszerek fejlesztésénél a minőség kulcskérdése egy minőségi kultúra kialakítása és annak

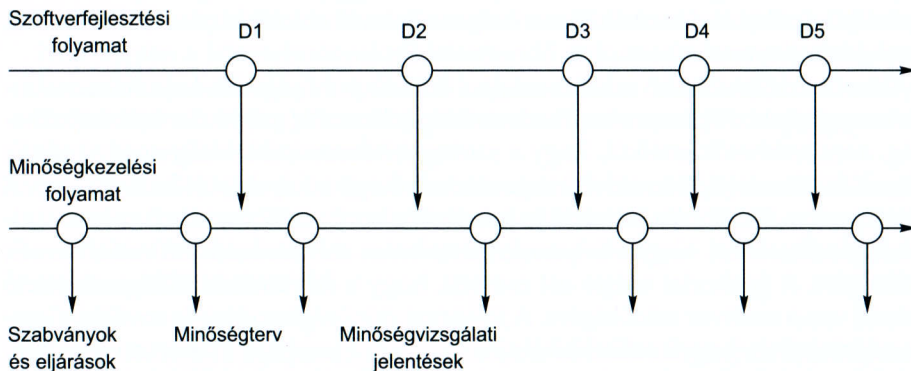
biztosítása, hogy a csapat minden tagja pozitív módon álljon a szoftverminőséghez.

A szoftverminőség kezelése nagy rendszerek esetén három fő tevékenységre osztható:

1. *Minőségbiztosítás.* Magas minőségű szoftverek előállítását eredményező szervezeti eljárások és szabványok rendszerének felállítása.
2. *Minőségtervezés.* A rendszerből a megfelelő eljárások és szabványok kiválasztása, és egy meghatározott szoftverprojekthez való igazítása.
3. *Minőség-ellenőrzés.* Azoknak a folyamatoknak a meghatározása és rendszerbe állítása, amelyek biztosítják, hogy szoftverfejlesztő csapat alkalmazza a projektminőségi eljárásokat és szabványokat.

A minőségkezelés független ellenőrzési lehetőséget nyújt a szoftverfejlesztési folyamathoz. A minőségkezelési folyamatot a szoftverkészítési folyamatban előállított átadandó termékeken hajtják végre, és vizsgálják, hogy ezek megfelelnek-e a szervezet szabványainak és céljainak (27.1. ábra). Mivel a minőségbiztosítást és a minőség-ellenőrzést független csapatoknak kell végezniük, ezért a folyamat tárgyilagosan véleményezhető, a problémákat és nehézségeket pedig jelenteni lehet a szervezet felső vezetésének.

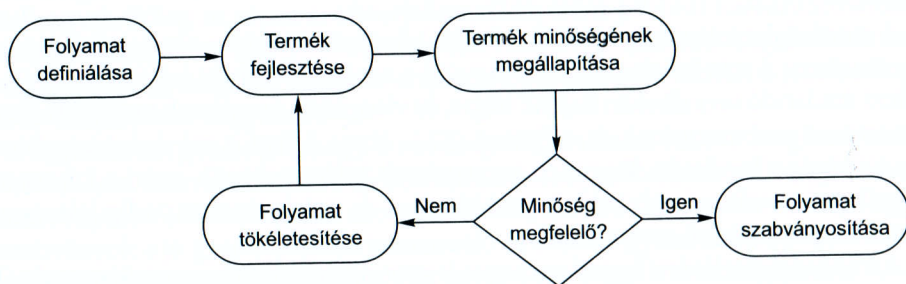
A minőségkezelésért független csapat legyen a felelős, akik a projektvezetőnél magasabb szintű vezetőség számára készítik a jelentéseket. Nekik egyetlen fejlesztői csoporttal sem szabad kapcsolatban állniuk, de vállalati szinten vállalniuk kell a felelősséget a minőség kezeléséért. Ez azért értelmes, mert a projektmenedzserek a projekt költségvetését és az ütemezését felügyelik, és ha probléma lép föl, akkor a termék minőségét megpróbálják csak olyan szinten kezelni, hogy az ne veszélyeztesse az ütemtervet. A független minőségkezelő csapat biztosítani tudja, hogy a szervezet minőségi céljai ne sérüljenek rövid távú pénzügyi vagy ütemezési megfontolások miatt.



27.1. ábra. Minőségkezelés és szoftverfejlesztés

27.1. A FOLYAMAT ÉS A TERMÉK MINŐSÉGE

A minőségkezelés egyik alapfeltevése, hogy a fejlesztési folyamat közvetlenül hat az átadott termék minőségére. Ezt a feltevést a gyáripari rendszerektől vették át, ahol a termék minősége és a gyártási folyamat között szoros összefüggés áll fent. Az automatizált tömegtermelő rendszereknél a folyamathoz hozzátartozik a konfigurálás, a gép beállítása és működése része a folyamatnak. Ha egyszer egy gép jól működik, akkor ebből természetes módon következik a termék jó minősége. A termék minősége mérhető és a folyamat addig változtatható, amíg a szükséges minőségi szintet el nem éri. A termékminőség elérésének ezt a folyamat alapú szemléletmódját a 27.2. ábra szemlélteti.



27.2. ábra. Folyamat alapú minőség

Az ipari termelésben egyértelmű a kapcsolat a folyamat és a termék minősége között, mert a folyamat viszonylag könnyen szabványosítható, és a felügyelete is egyszerű. Ha a gyári rendszereket egyszer beállították, akkor kiváló minőségű termékek előállításához csak újra el kell indítani őket. A szoftvereket azonban nem gyártják, hanem tervezik. Mivel a szoftverfejlesztés inkább alkotó, mintsem mechanikus folyamat, erősen hatnak rá az egyéni képességek és tapasztalatok, de a termékminőséget a felhasznált folyamatától függetlenül külső tényezők is befolyásolják (például a termék újszerűsége, a termék mielőbbi piacra dobásának kereskedelmi nyomása).

A szoftverfejlesztésben a folyamat és a termék minőségének kapcsolata azonban bonyolultabb. Bizonyos szoftverminőség-jellemzők, például a karbantarthatóság, nem mérhetők anélkül, hogy a szoftvert huzamosabb ideig ne használták volna. Következésképpen nehéz megmondani, hogy a folyamat jellemzői hogyan hatnak ezekre. Továbbá a tervezés és kreativitás szoftverfolyamatbeli szerepe miatt megjósolhatatlan, hogy a folyamat változtatása milyen hatással lesz a termék minőségére. A gyakorlat mégis azt mutatja, hogy a folyamat minősége alapvető hatással van a szoftver minőségére. A folyamat minőségkezelése és továbbfejlesztése oda vezethet, hogy kevesebb hiányosság forduljon elő az átadott szoftverben.

A folyamatminőség kezelése a következőket jelenti:

1. Folyamat szabványok meghatározása (például mikor és hogyan ajánlott felülvizsgálatokat tartani).

2. A fejlesztési folyamat felügyelete annak biztosítására, hogy minden a szabványok szerint történik.
3. Jelentéskészítés a szoftverfolyamatról a projektvezetőségnek és a szoftver vásárlójának.

A folyamatra alapozott minőségbiztosításnak egyik veszélye, hogy nem biztos, hogy az előírt folyamat alkalmazható a fejlesztett szoftver típusához. A folyamatminőségi szabványok előírhatják például, hogy az implementálás nem kezdhető el a specifikáció elkészítése és jóváhagyása előtt, azonban néhány rendszernél szükség lehet prototípus-készítésre, amihez viszont implementálni kell a programot a teljes specifikáció hiányában. Ilyen esetekben a minőségügyi csapat, mivel a minősége úgysem felügyelhető, javasolhatja a prototípus-készítés elhagyását, de ekkor a felsőbb vezetésnek közbe kell lépnie, hogy biztosítsa: a minőségi folyamat nem hátráltatja, hanem támogatja a termék fejlesztését.

27.2. A MINŐSÉGBIZTOSÍTÁS ÉS A SZABVÁNYOK

A minőségbiztosítás az a folyamat, amikor meghatározzák, hogy a szoftver minősége hogyan valósítható meg, és hogy a fejlesztő szervezet hogyan tudja megállapítani, hogy a szoftver már elérte a minőség szükséges szintjét. Ahogy már javasoltam, a minőségbiztosítási folyamat első lépése a szoftverfejlesztési folyamatra vagy a szoftvertermékre alkalmazandó szabványok meghatározása vagy kiválasztása legyen. A minőségbiztosítási folyamat részeként kiválaszthatók és beszerezhetők olyan eszközök és módszerek, amelyek támogathatják ezeket a szabványokat.

A minőségbiztosítási folyamat részeként kétféle szabvány állítható fel:

1. *Termékszabványok.* A fejlesztett szoftvertermékre alkalmazott szabványok. Ezek közé tartoznak a dokumentumokra vonatkozó szabványok (például milyen legyen a követelményekről készítendő dokumentum szerkezete), a dokumentációs szabványok (például hogyan nézzen ki az objektumosztályok definíciójának fejléce) és a kódolási szabványok, amelyek a programozási nyelv használatát határozzák meg.
2. *Folyamatszabványok.* A szoftverfejlesztés alatt követendő folyamatokat meghatározó szabványok. Vonatkozhatnak a specifikáció definíciójára, a tervezési és validálási folyamatokra és mindezen folyamatok során létrehozandó dokumentumok leírására.

Ahogy azt a 27.1. alfejezetben kifejtettem, a termékszabványok és a folyamatszabványok között nagyon szoros a kapcsolat. A termékszabványokat a szoftverfolyamat kimenetére alkalmazzák, és sok esetben vannak olyan speciális tevékenységek a folyamatszabványok között, amelyek biztosítják a termékszabványok betartását.

Számos oka van annak, miért olyan fontosak a szoftverszabványok:

1. Egybegyűjtik a legjobb vagy a legkevésbé megfelelő gyakorlatokat, amelyekre vonatkozó ismeretek gyakran csak nagyon sok próbálgatás és kudarc árán szerezhetők meg. Szabványba foglalásukkal elkerülhető a régi hibák újbóli elkövetése, és nem vesznek el a szervezet számára értékes tapasztalatok.
2. Olyan keretrendszert adnak, amely köré implementálható a minőségbiztosítási folyamat. Ha a legjobb gyakorlatok már szabványosítva vannak, akkor a minőség-ellenőrzés egyszerűen a szabványok megfelelő kiválasztását és követését biztosítja.
3. Elősegítik a folytonosságot ott, ahol valaki másnak kell folytatnia a megkezdett munkát. A szabványokkal biztosítani lehet, hogy egy szervezeten belül a mérnökök ugyanazt a gyakorlatot alkalmazzák, így ha új munkába kezdenek bele, csökken a tanulásba fektetett energia mennyisége.

A szoftvertervezési projektek szabványainak kifejlesztése bonyolult és időigényes folyamat. A szabványokat olyan nemzeti és nemzetközi testületek hozzák létre, például az Egyesült Államok Nemzetvédelmi Minisztériuma, az ANSI, a BSI, a NATO és az IEEE; szabványaik különféle projektekre alkalmazható általános szabványok. A NATO-hoz vagy más védelmi szervezetekhez hasonló szervezetek szoftverszerződéseikben saját szabványaik betartását is megkövetelhetik.

A nemzeti és nemzetközi szabványokat úgy fejlesztették ki, hogy azok terjedjen ki a szoftvertervezési terminológiára, a programozási nyelvekre (például Java vagy C++), a jelölésrendszerre (például a folyamatábrák szimbólumai), a szoftverkövetelmények származtatására és írására használt eljárásokra, a minőségbiztosítási eljárásokra, valamint a szoftver-ellenőrzési és -validálási folyamatokra (IEEE, 2003).

Szabványokat fejlesztő minőségbiztosítási csapatoknak általában ajánlott a szervezeti szabványokat a nemzeti és nemzetközi szabványok alapján kialakítaniuk, és ezekből kiindulva olyan „szabványkézikönyvet” kellene összeállítaniuk, amely a szervezetre szabott szabványokat definiálja. Egy ilyen kézikönyvben például a 27.3. ábrán megadott szabványok szerepelhetnek.

Termékszabványok	Folyamatszabványok
A terv áttekintésének űrlapjai	A terv áttekintésének irányítása
A követelményeket leíró dokumentumok szerkezete	A dokumentumok ellenőrzése
Az eljárásfej formátuma	Verziókibocsátási folyamat
Java programozói stílus	A projektterv jóváhagyási folyamata
A projektterv formátuma	A változásvezérlő folyamat
A változtatási kérelmek űrlapjai	A tesztek rögzítésének folyamata

27.3. ábra. Termék- és folyamatszabványok

A szoftvermérnökök néha úgy vélik, hogy a szabványok bürokratikusak, és nem tartoznak hozzá a szoftverfejlesztés szakmai részéhez. Erre különösen akkor van esély, ha a szabványok betartása unalmas úrlaptöltötéssel és munkaregisztrációval jár. És bár rendszerint egyetértenek abban, hogy általánosságban szükség van szabványokra, de gyakran jó okokat tudnak felhozni, hogy a szabványok miért nem felelnek meg igazán egy adott projekthez.

A fenti problémák elkerülésére a szabványokat előíró minőségi vezetők számára biztosítani kell a szükséges erőforrásokat, nekik pedig a következő intézkedéseket kellene megtenniük:

1. Szoftvermérnökök bevonása a termékszabványok fejlesztésébe. A szoftvermérnököknek meg kellene érteniük a szabványfejlesztések okát, és el kellene kötelezniük magukat e szabványok mellett. A szabványok dokumentációjának nem egyszerűen a követendő szabványt kell ismertetnie, hanem meg is kell magyaráznia, hogy a szabványosításnál miért az adott döntések születtek meg.
2. A szabványok rendszeres felülvizsgálata és módosítása a technológiai változások követésének céljából. A szervezetek hajlamosak a kifejlesztett szabványokat gondosan őrizgetni a szervezeti szabványokat tartalmazó kézikönyvben, és gyakran nem szívesen vállalkoznak megváltoztatásukra. Fontos, hogy a szervezetnek legyen szabványokat tartalmazó kézikönyve, de ezt a körülmények és a technológiák változásával párhuzamosan fokozatosan alakítani kell.
3. Szoftvereszközök biztosítása a szabványok támogatására minden lehetséges helyen. Sokat panaszkodnak az adminisztrációs szabványokra, mert implementálásuk unalmas munkával jár. Ha rendelkezésre állnak támogatóeszközök, a szabvány kevesebb munkával is kifejleszthető.

A folyamatszabványok nehézségeket okozhatnak, ha a fejlesztőcsapatnak kivitelezhetetlen folyamatot írnak elő. Különböző szoftverek különböző fejlesztési folyamatot igényelnek. Így nem lehet előírni az általános használatot, ha az nem felel meg a projektnek vagy a projektcsapatnak. A projektvezetőket fel kell jogsítani arra, hogy módosíthassák a folyamatszabványokat az adott körülmények függvényében, de a termék minőségére és a leszállítás utáni folyamatra vonatkozó szabványokat csak alapos megfontolás után szabad megváltoztatni.

A nem megfelelő szabványok alkalmazása elkerülhető, ha a projektvezető és a minőségi vezető körültekintően tervezi meg a minőséget. Az ő hatáskörükbe tartozik annak eldöntése, hogy a kézikönyvben lévő szabványok közül melyeket alkalmazzák változtatás nélkül, melyek szorulnak módosításra, és melyeket lehet figyelmen kívül hagyni. Bizonyos projektkövetelményekből adódóan új szabványok létrehozására is szükség lehet (például a korábbi projektekben nem használt formális specifikációkhoz kell új szabványokat készíteni). Ahogy a csapat egyre gyakorlottabb lesz bennük, megtervezhetjük ezen új szabványok módosítását és kiterjesztését.

27.2.1. ISO 9000

A minőségkezelési rendszereknek az egyik, minden iparágában használható nemzetközi szabványa az ISO 9000, amely a gyáripartól kezdve a szolgáltató-ágazatig, különféle szervezetekre alkalmazható szabványok együttese. Szabványai közül az ISO 9001 szabvány a legáltalánosabb, ez a termékek tervezését, fejlesztését és karbantartását végző szervezetek minőségi folyamataival foglalkozó szervezetekre vonatkozik. Az ISO 9000 szabványt szoftverfejlesztéshez egy kiegészítő dokumentum (ISO 9000-3) segítségével lehet felhasználni. Az ISO 9000 szabvány leírásáról számos könyv kapható (Johnson, 1993; Oskarsson és Glass, 1995; Peach, 1996; Bamford és Deibler, 2003).

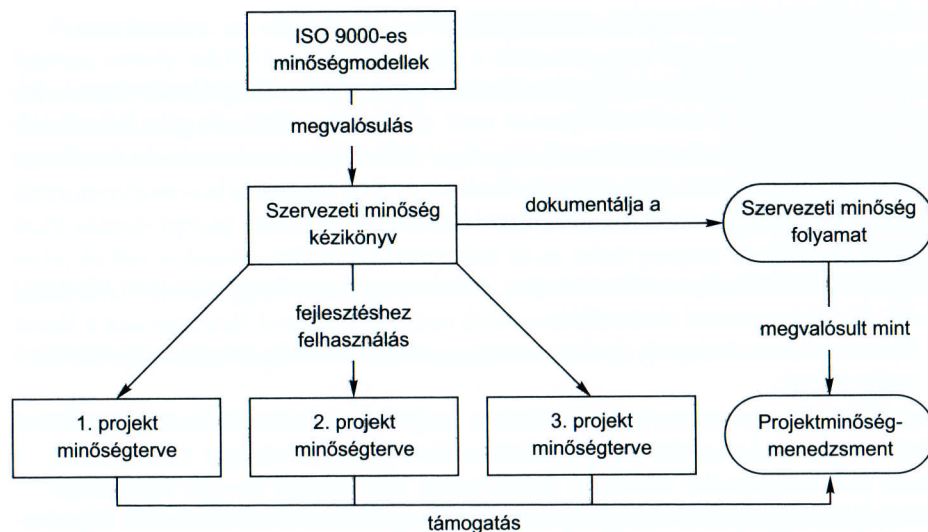
Az ISO 9001 szabvány nem specifikusan a szoftverfejlesztést célozza, de összefoglalja azokat az általános elveket, amelyek alkalmazhatók a szoftverre. Az ISO 9001 szabvány több szemszögből is leírja a folyamatot, és meghatározza a szervezeteknél alkalmazandó szabványokat és eljárásokat. Ezeket kell dokumentálni a szervezet minőségi kézikönyvében. A folyamat definíciójának tartalmaznia kell a dokumentáció leírásait, amelyek bizonyítják, hogy a definiált folyamatokat betartották a termék fejlesztése során.

Az ISO 9001 nem ad meg minőségi folyamatot, amit használni lehetne. Ténylegesen nem korlátozza a bármely szervezetnél bármikor használható folyamatokat. Ez az iparban nagymértékű hajlékonyságot biztosít, ez pedig azt jelenti, hogy a kis vállalatok bürokrácia mentes folyamatokkal rendelkezhetnek, és mégis megfelelnek az ISO 9000-nek. Ugyanakkor ez a hajlékonyság azt is jelenti, hogy nem tételezhetünk föl semmit az ISO 9000-nek megfelelő vállalatok folyamatainak hasonlóságáról vagy különbözőségéről.

Az ISO 9001 alkalmazási területei a 27.4. ábrán láthatók. Itt terjedelmi okok miatt nem foglalkozom a szabvánnyal teljes mélységében, viszont Ince (Ince, 1994), valamint Oskarsson és Glass (Oskarsson és Glass, 1995) sokkal részlete-

Vezetési felelősségek	Minőségi rendszer
Rendhagyó termékek ellenőrzése	A tervezés ellenőrzése
Kezelés, tárolás, csomagolás és szállítás	Beszerzés
Beszerzett termékek	A termék azonosítása és nyomon követhetősége
Folyamat-ellenőrzés	Ellenőrzés és tesztelés
A berendezések ellenőrzése és tesztelése	Az ellenőrzés és a tesztelés állapota
A szerződések felülvizsgálata	Javítási művelet
A dokumentálás ellenőrzése	Minőségi jelentések
Belső minőségi felülvizsgálatok	Képzés
Szervizelés	Statisztikai technikák

27.4. ábra. Az ISO 9001 minőségbiztosítási modell alkalmazási területei



27.5. ábra. Az ISO 9000 és a minőségkezelés

sebb leírást adnak arról, hogy a szabvány hogyan alkalmazható szoftverek minőségkezelési folyamatainak fejlesztésére. Az ISO 9000, a minőségi kézikönyv és az egyedi projektminőségi tervek közötti kapcsolatot a 27.5. ábra mutatja. Ezt az ábrát Ince könyvének (Ince, 1994) egy modelljéből vettem.

A szervezetek a minőségi folyamatokat leíró minőségi kézikönyvbe jegyzik fel minőségbiztosítási eljárásaikat. Néhány országban olyan bizottságokat hoztak létre, amelyek tanúsítják, hogy a minőségi kézikönyv szerinti minőségi folyamatok megfelelnek az ISO 9001 szabványnak. A vásárlók egyre gyakrabban kérnek szállítóiktól ISO 9000 tanúsítványt mint annak mutatóját, mennyire veszik komolyan a szállítók a minőséget.

Sokan azt hiszik, hogy az ISO 9000 tanúsítvány azt jelenti, hogy a tanúsított vállalatok által előállított szoftverek minősége jobb, mint a nem tanúsítottaké. Bizonyosan nem ez a helyzet. Az ISO 9000 szabvány egyszerűen csak arra vonatkozik, hogy a vállalatnál használatban van definiált folyamat és van hozzá tartozó dokumentáció, amely bizonyítja, hogy ezt a folyamatot alkalmazzák. De azt nem biztosítja, hogy ezek a folyamatok a legjobb gyakorlatot tükrözik, vagy hogy a minőség szavatolt.

Lehetséges, hogy a vállalat olyan terméktesztelő eljárást definiál (mondjuk), amely nem eredményez teljes szoftvertesztelést. Attól kezdve, hogy ezeket az eljárásokat alkalmazza, és ezt dokumentálja, a vállalat megfelel az ISO 9001 szabványnak. Bár ez a szituáció valószínűtlen, az nem vitás, hogy egyes vállalatok szabványai meglehetősen gyengék és kevésbé befolyásolják a szoftver tényleges minőségét.

27.2.2. Dokumentációs szabványok

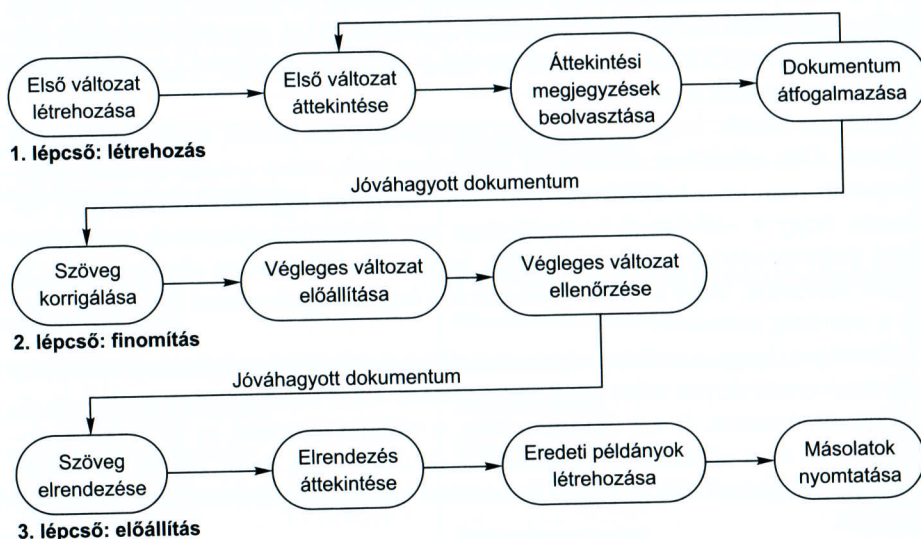
A szoftverfolyamat során különösen fontosak a dokumentációs szabványok, hiszen a szoftver és a szoftverfolyamat csak a dokumentáció alapján követhető. A szabványosított dokumentumok egységes külalakja, szerkezete és minősége miatt könnyebb olvasni és megérteni őket.

Háromféle dokumentációs szabvány létezik:

1. *A dokumentációs folyamat szabványai.* A dokumentum elkészítése alatt alkalmazandó folyamatokat definiálják.
2. *Dokumentumszabványok.* A dokumentum szerkezetét és megjelenítését előíró szabványok.
3. *A dokumentumcsere szabványai.* Ezek a szabványok biztosítják, hogy a dokumentum minden elektronikus másolata kompatibilis legyen.

A folyamatszabványok a dokumentumok előállítására alkalmazott folyamatokat határozzák meg, azaz a dokumentum fejlesztése alatt használt eljárásokat és a dokumentumkészítés szoftvereszközeit, de a dokumentumok magas minőségét biztosító ellenőrző és finomítási eljárásokat is definiálni kell.

A dokumentációs folyamat minőségi szabványai legyenek rugalmasak, és legyenek képesek mindenféle típusú dokumentum kezelésére. Munkairatok és feljegyzések esetében nincs szükség részletes minőség-ellenőrzésre, azonban ajánlott formális minőségi folyamatokat alkalmazni a későbbi fejlesztéshez szükséges formális dokumentumoknál vagy a vásárlónak átadandó dokumentumoknál. Egy ilyen lehetséges folyamat modellje a 27.6. ábrán látható.



27.6. ábra. Egy minőség-ellenőrzést magában foglaló, dokumentumot előállító folyamat

A szerkesztés, az ellenőrzés, az átvizsgálás és az újraserkesztés iteratív folyamat, amely addig ismétlődik, amíg a dokumentum elfogadható minőségűvé nem válik. Az elfogadható minőség értékelése függ a dokumentum típusától és a dokumentum potenciális olvasóitól.

A dokumentumszabványokat a szoftverfejlesztés folyamán készített minden dokumentumra alkalmazni kell. A dokumentumok stílusának, külsejének, valamint azonos típusú dokumentumoknál a szerkezetüknek is egységesnek kell lenniük, és bár a dokumentumszabványokat is az adott projekt kívánalmaihoz kell igazítani, hasznos gyakorlat ugyanazt a „vállalati stílust” alkalmazni az ugyanannál a szervezetnél készített minden dokumentumra.

Néhány példa kifejleszthető dokumentumszabványokra:

1. *Dokumentumazonosítási szabványok.* Mivel a nagy rendszereket fejlesztő projektben több ezer dokumentumot is előállíthatnak, ezért minden egyes dokumentumot egyedi módon kell azonosítani. Formális dokumentumok esetén ez az azonosító lehet a konfigurációkezelő program által meghatározott formális azonosító, informális dokumentumoknál pedig a projektvezetőnek kell definiálnia a dokumentumazonosító formáját.
2. *A dokumentum szerkezetére vonatkozó szabványok.* A szoftverprojekt során előállított minden egyes dokumentumosztálynak szabványos szerkezetűnek kell lennie. A szerkezetre vonatkozó szabványok határozzák meg, milyen szakaszokból álljon a dokumentum, és rögzítik az oldalszámozásra, az oldal fejléc- és lábléc-információira, valamint a fejezetek és alfejezetek számozására vonatkozó előírásokat.
3. *A dokumentum kinézetére vonatkozó szabványok.* Ezek a szabványok határozzák meg a „vállalati stílust”, és jelentősen hozzájárulnak a dokumentumok konzisztenciájához. Definálják például a dokumentumban használt betűtípusokat és stílusokat, a logók és vállalatnevek használatát vagy a dokumentum szerkezetét kiemelő színek alkalmazását.
4. *A dokumentum aktualizálására vonatkozó szabványok.* Mivel a dokumentumok a rendszer változásainak követése közben alakulnak, módosításukat konzisztens módon kellene elvégezni. A dokumentumok frissebb verzióit jelezhetik különböző színű fedőlapok, és a megváltoztatott vagy újonnan beleírt bekezdések mellé húzható egy vonal a margóra. Ugyanakkor nem javaslok az általánosan használt szövegszerkesztők által támogatott változáskövetők használatát. Ha már kettőnél több szerző van, a változáskövetés már inkább zavar és nem segít.

Fontosak a dokumentumcserére vonatkozó szabványok, mivel a dokumentumoknak az elektronikus másolatait cserélgetik. Ezek a szabványok teszik lehetővé a dokumentumok elektronikus átvitelét, és az eredeti formájukban való újbóli létrehozásukat.

Feltéve, hogy a folyamatszabványok szabványos eszközök használatát is tartalmazzák, a dokumentumcserére vonatkozó szabványok rendelkeznek ezen

eszközök használatára vonatkozó előírásokról is. Dokumentumcserére vonatkozó szabvány lehet például az egységes stíluslap, szövegszerkesztők esetén, vagy a dokumentummakrók használatának korlátozása a vírusfertőzés elkerülése érdekében. A különböző nyomtatási és megjelenítési lehetőségek miatt a cserére vonatkozó szabványok korlátozhatják a használt betűtípusokat és szövegstílusokat is.

27.3. A MINŐSÉGTERVEZÉS

A minőségtervezési folyamat eredménye a projekt minőségi terve. Az elvárt termékminőséget és mérésének módját a minőségi tervnek kell rögzítenie, következőképpen a minőségi terv definiálja, hogy ténylegesen mi is nevezhető „magas minőségű” szoftvernek. Ilyen meghatározás hiányában megeshetne, hogy az egyes mérnökök egymásnak teljesen ellentmondó munkamódszerekkel dolgoznak, és így különböző termékjellemzőket optimalizálnának.

A minőségi tervben ki kell választani az adott termékre vagy fejlesztési folyamatra alkalmazható szervezeti szabványokat, és ha új módszereket vagy eszközöket is használnak a projektben, akkor új szabványok definiálására is sor kerülhet. Humphrey (Humphrey, 1989) a szoftvertervezésről írt klasszikus könyvében egy vázlatos szerkezetet javasol a minőségi tervekhez, amely a következő pontokat tartalmazza:

1. *A termék bemutatása.* A termék leírása, a termék megcélzott piaca és a termékhez fűződő minőségi elvárások.
2. *A terméktervek.* A kritikus kibocsátási időpontok és a termékért vállalt kötelezettségek, valamint a terjesztésre és a termék szervizelésére vonatkozó tervek.
3. *A folyamatok leírása.* A termék fejlesztése és menedzselése közben használt fejlesztési és szolgáltatási folyamatok leírása.
4. *A minőségi célok.* A termékre vonatkozó minőségi célok és tervek, többek között a kritikus termékminőségi jellemzők azonosítása és indoklása.
5. *A kockázatok és a kockázatkezelés.* Azok a fő kockázatok, amelyek befolyásolhatják a termék minőségét, és ezeket a kockázatokat megcélzó tevékenységek.

A minőségi tervek különböznek részleteikben, attól függően, hogy az éppen fejlesztett rendszer mekkora méretű és milyen típusú. Ugyanakkor a minőségi tervek írásakor törekedni kell arra, hogy a terv a lehető legrövidebb legyen. Ha a dokumentum túl hosszú, az emberek nem fogják elolvasni, és így a minőségi tervek nem érik el a céljukat.

A minőségtervezési folyamat során a szoftverminőség lehetséges jellemzőinek széles skáláját kell figyelembe venni (27.7. ábra). Általában egyetlen rendszeren sem lehet e jellemzők mindegyikét egyidejűleg optimalizálni, éppen ezért a minőségtervezésnek egyik kritikus pontja, hogy kiválasszák a kritikus minőségi jellemzőket, és megtervezzék ezek megvalósítását. A minőségi tervnek meg kell határoznia a fejlesztés alatt álló termék legfontosabb minőségi jellemzőit. Ha a

Biztonságosság	Érthetőség	Hordozhatóság
Biztonság	Tesztelhetőség	Felhasználhatóság
Megbízhatóság	Alkalmazhatóság	Újrafelhasználhatóság
Rugalmasság	Modularitás	Hatékonyság
Robusztusság	Komplexitás	Megtanulhatóság

27.7. ábra. A szoftver minőségi jellemzői

jellemzők közül a hatékonyságot emelik ki, a többi jellemzőt feláldozzák ennek érdekében, és ha mindezt a tervben is rögzítették, akkor a fejlesztésen dolgozó mérnökök együtt tudnak működni ennek megvalósításában. A tervnek definiálnia kell a minőségértékelési folyamatot is, amelynek valamilyen minőségi jellemző (például a karbantarthatóság vagy a robusztusság) jelenlétét kell szabványos módon felmérnie a termékben.

27.4. A MINŐSÉG-ELLENŐRZÉS

A minőség-ellenőrzés a szoftverfejlesztési folyamat minőségbiztosítási folyamatnak végrehajtását és a szabványok betartásának vizsgálatát jelenti. Ahogyan azt korábban a fejezetben kifejtettem (27.1. ábra), a minőség-ellenőrzési folyamat során a szoftverfolyamat átadásra kerülő részeit ellenőrzi, hogy megfelelnek-e a projekt szabványainak.

A minőség-ellenőrzés két, egymást kiegészítő szempontból közelíthető meg:

1. Minőségi felülvizsgálatnál magát a szoftvert, az előállításánál használt dokumentációt és folyamatokat emberek egy csoportja nézi át. A felülvizsgálat a felelős annak ellenőrzéséért, hogy követték-e a projektszabványokat, a szoftver, a dokumentumok megfelelnek-e ezeknek a szabványoknak. A szabványtól való eltéréseket feljegyzik, és felhívják rá a projektvezető figyelmét.
2. Automatikus szoftverértékelésnél a szoftvert és az előállított dokumentumokat valamilyen program dolgozza fel, és vizsgálja az adott fejlesztői projektre alkalmazható szabványoknak való megfelelést. Az automatikus értékelésnek része lehet néhány szoftverjellemző mennyiségi mérése is. A szoftverméréssel és a metrikákkal a 27.5. alfejezetben foglalkozom.

27.4.1. Minőségi felülvizsgálat

A folyamatok vagy termékek minőségének validálására a leghatékonyabb körben használt módszer a felülvizsgálat. A felülvizsgálatot emberek egy csoportja végzi el, akik azért vizsgálják át a szoftverfolyamat, a rendszer és a hozzá tartozó dokumentáció egészét vagy egy részét, hogy rejtett problémákat fedezzenek fel. A fe-

lülvizsgálatból levont következtetéseket hivatalosan is feljegyzik, és átadják a felülvizsgálat szerzőjének vagy a feltárt problémák kijavításáért felelős személynek.

A 27.8. ábra a felülvizsgálat több különböző típusát mutatja be. A programok felülvizsgálatáról a 22. fejezetben, a folyamatok felülvizsgálatáról pedig az 5. fejezetben tárgyalt menedzselési folyamat részeként volt szó, itt a felülvizsgálatokkal mint a minőségkezelési folyamatok részével foglalkozunk. A különböző felülvizsgálati folyamatoknak nagyon sok közös vonása van, a felülvizsgálat bevezetésének folyamatát már leírtam a 22. fejezetben.

A felülvizsgálat típusa	Legfőbb célja
A terv vagy a program vizsgálata	Aprólékos hibakeresés a követelményekben, a tervben vagy a kódban. A felülvizsgálatot a lehetséges hibák listája alapján kellene elvégezni.
Az előrehaladás felülvizsgálata	Információt nyújt a vezetőségnek a projekt általános előrehaladásáról, de egyben a folyamat és a termék felülvizsgálata is, továbbá kapcsolódik a költségekhez, a tervekhez és az ütemezésekhez is.
A minőség felülvizsgálata	A termékkomponensek vagy a dokumentáció technikai elemzése a komponensek terve, kódja vagy dokumentációja és a specifikáció közötti eltérések felfedése és a megállapodás szerinti szabványok betartásnak biztosítása céljából.

27.8. ábra. A felülvizsgálat típusai

A felülvizsgálatot végző csapat feladata az, hogy hibákat és ellentmondásokat fedezzenek fel, és felhívják rá a tervezőnek vagy a dokumentum szerzőjének a figyelmét. Felülvizsgálni nemcsak a specifikációkat, a terveket és a kódot lehet, hanem más dokumentumokat is (például folyamatmodelleket, teszterveket, konfigurációkezelési eljárásokat, folyamatszabványokat és felhasználói kézikönyvet).

A felülvizsgálatot végző csapatba azokat a projekttagokat is ajánlott bevenni, akik hatékonyan közre tudnak működni a felülvizsgálatban. Például ha egy alrendszer felülvizsgálata folyik, akkor a kapcsolódó alrendszerek tervezőit is érdemes bevenni a csapatba, mert ők jól ismerhetik az alrendszer azon interfészeit, amelyeket, ha az alrendszert csak önmagában vizsgálnák, esetleg kihagynának.

A felülvizsgálatot végző csapatnak legyen egy 3-4, vezető felülvizsgálónak választott személyből álló magja. Ezek egyikének vezető tervezőnek kellene lennie, aki vállalni tudja a felelősséget a jelentősebb technikai döntésekért. A vezető felülvizsgálók más tagokat is felkérhetnek a projektből felülvizsgálatra, akiknek nem a teljes dokumentációt, hanem csak a munkájukat érintő részeket kell átnézniük, vagy a másik lehetséges eset szerint a felülvizsgálatot végző csapat tagjai körbeadják a vizsgált dokumentációt, és más projekttagokat kérnek fel észrevételeik feljegyzésére.

A felülvizsgálatra szánt dokumentumokat jóval a felülvizsgálat előtt ki kell adni, hogy a vizsgálatot végzőknek legyen elég ideje elolvasni és megérteni őket. Az okozott időkiesés ugyan megakaszthatja a fejlesztési folyamatot, viszont ha a felülvizsgálat előtt a csapat nem érti meg rendesen a dokumentumokat, akkor a felülvizsgálat eredménytelen lesz.

A felülvizsgálatnak nem szabad sokáig tartania (legfeljebb kétórás legyen). A vizsgált dokumentum szerzőjének a csapattal együtt végig kell haladnia a dokumentumon. Legyen egy kijelölt csapattag, aki hivatalosan feljegyez minden felülvizsgálati döntést, egy másik pedig a felülvizsgálat elnöklésére, akinek vállalnia kell a felelősséget annak biztosításáért is, hogy minden írott észrevételt megvitatnak. A felülvizsgálat befejeztével feljegyzik a tevékenységeket, az észrevételeket és tevékenységeket tartalmazó űrlapot aláírattják a tervezővel és a felülvizsgálat elnökével, majd ezeket a hivatalos projektdokumentáció részeként az irattárba helyezik. A szükséges változtatások elvégzésének biztosításáért az elnök a felelős. Ha nagyobb változtatások szükségesek, akkor újabb felülvizsgálatot lehet elrendelni.

27.5. A SZOFTVER MÉRÉSE ÉS A METRIKÁK

A minőségi felülvizsgálatok drágák és időigényesek, és szükségszerűen hátráltatják a szoftverrendszer elkészültét. Ideális esetben lehetséges lenne a felülvizsgálati folyamatot úgy kialakítani, hogy eszközöket használunk a szoftvertervezési és programozási folyamathoz, és a szoftverminőséget automatikusan értékeljük. Ez az értékelés ellenőrizné, hogy a szoftver rendelkezik-e már a megkövetelt minőségi küszöbvel, és ha nem, akkor kiemelné a szoftver azon területeit, ahová a felülvizsgálatot fókuszálni kell.

A szoftvermérés azzal foglalkozik, hogyan lehet a szoftvertermék vagy a szoftverfolyamat valamely jellemzőjéből numerikus értéket előállítani. Ezen értékek egymással és az egész szervezetre alkalmazott szabványokkal való összehasonlításból levonhatók bizonyos következtetések a szoftver vagy a szoftverfolyamat minőségére vonatkozóan. Például tegyük fel, hogy egy szervezet új szoftvertesztelő eszköz bevezetését tervezi. Az eszköz bevezetése előtt feljegyzik az adott idő alatt felfedezett szoftverhibák számát, majd az eszköz bevezetése után megismétlik a folyamatot. Ha az eszköz bevezetése után ugyanannyi idő alatt több hibát vesznek észre, akkor az eszköz hasznos támogatója lehet a szoftvervalidálási folyamatnak.

Két olyan terület van, ahol a szoftvertermékmérések használhatók:

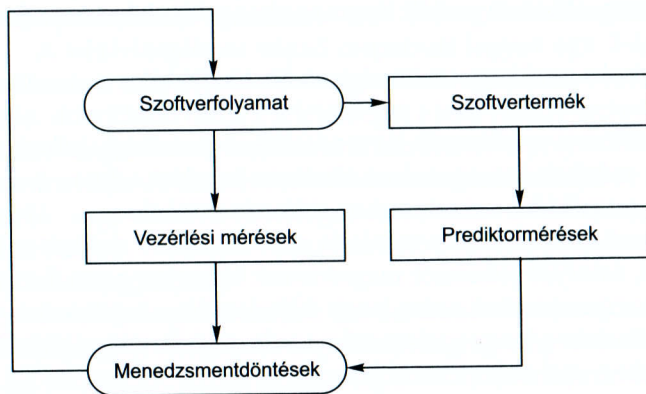
1. *Általános előrejelzés készítése a rendszerről.* A rendszerkomponensek jellemzőinek mérésével, és a mérések összegzésével általános becslést adhatunk a rendszer jellemzőire, mint például a rendszerben előforduló hibákra.
2. *A rendellenes komponensek azonosítása.* A mérések azonosíthatják azokat az egyedi komponenseket, amelyek jellemzői megsértenek bizonyos normákat. Például mérhetjük a komponenseket azért, hogy felfedezzük a legösszetettebbeket, és ezután (feltételezve, hogy ezek tartalmazzák a hibák többségét) a felülvizsgálati folyamatban ezekre koncentrálhatunk.

Szoftvermetrikának neveznek bármely olyan típusú mérést, amely egy szoftverrendszerhez, szoftverfolyamathoz vagy az ezekhez kapcsolódó dokumentációhoz fűződik (például a termék méretének kifejezése a kódsorok számában, az írott szövegben a szakaszok olvashatóságát mérő Fog index (Gunning, 1962), az átadott szoftvertermékben talált hibák száma vagy a rendszerkomponensek fejlesztéséhez szükséges embernapok száma).

Számos nagyvállalat, mint például a Hewlett–Packard (Grady, 1993) és az AT&T (Barnard és Price, 1994) és a Nokia (Kilpi, 2001) vezetett be mérési programokat, és használ összegyűjtött metrikákat minőségkezelési folyamataihoz. A hangsúlyt főleg a programhibákat mérő metrikák gyűjtésére, valamint a verifikáló és validáló folyamatokra helyezik. A mérési programok gyáripárba való bevezetéséről Offen és Jeffrey (Offen és Jeffrey, 1997), valamint Hall és Fenton (Hall és Fenton, 1997) írt, az ami-kézikönyv (application of metrics in industry – metrikák alkalmazása az iparban) (Pulford és mások, 1996) pedig részletes tanácsokat ad a mérésekhez és a folyamatok továbbfejlesztésében való felhasználásukhoz.

A szisztematikus szoftvermérés alkalmazása azonban viszonylag még mindig ritka, és idegenkednek is a mérések bevezetésétől, mert nem ismert, milyen előnyökkel járhat. Ennek egyik oka az, hogy sok szervezetben a szoftverfolyamatok szervezettsége még mindig alacsony fokú, és a szervezetek nem is elég érettek arra, hogy hasznát vegyék a méréseknek. Másik oka a mérések szabványosításának a hiánya, ami miatt korlátozott mértékű az adatgyűjtés és az elemzés támogatása. Sok vállalat addig nem is lesz felkészült a mérések bevezetésére, amíg nem állnak rendelkezésre mérésekre vonatkozó szabványok és eszközök.

A metrikák lehetnek vezérlési vagy prediktormetrikák. Mind a vezérlési, mind a prediktormetrikák befolyásolják a vezetői döntéshozatalt a 27.9. ábrán látható módon. A vezérlési metrikák általában a szoftverfolyamattal kapcsolatosak, míg a prediktormetrikák a szoftvertermékkel. Vezérlési vagy folyamatmetrika például a feljegyzett hibák kijavításához szükséges átlagos idő és munka, prediktormetrikára pedig a modul ciklomatikus komplexitása, a program azonosítóinak



27.9. ábra. Prediktor- és vezérlési metrikák

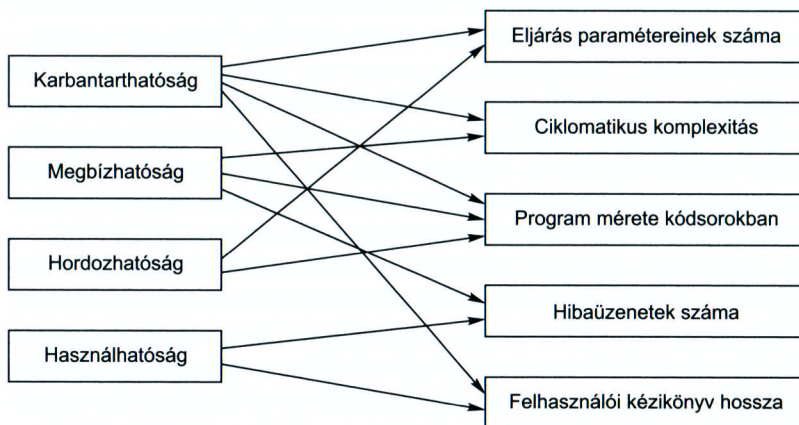
átlagos hossza vagy a tervezett objektumok műveleteiben szereplő attribútumok száma lehet példa. A folyamatmetrikákra és a folyamatok továbbfejlesztésében betöltött szerepükre a 28. fejezetben térek vissza, ebben a fejezetben a szoftvertermék minőségének előrejelzésére használt mérésekkel foglalkozom.

A szoftver minőségi jellemzőit gyakran lehetetlen közvetlenül lemérni. A külső jellemzők, mint például a karbantarthatóság, a bonyolultság és érthetőség, arra vonatkoznak, hogy a fejlesztők és a felhasználók hogyan látják a szoftvert. Több különböző tényező hat rájuk, és nem egyszerű a mérésük. Helyettük a szoftver valamely belső jellemzőjét (például a méretet) kell mérni, és fel kell tenni, hogy valamilyen kapcsolat van aközött, amit mérni tudunk és amit tudni szeretnénk. Ideális esetben a belső és külső szoftverjellemzők között világos és validált kapcsolatnak kell lennie.

A 27.10. ábrán látható néhány, esetlegesen érdeklődésre számot tartó külső minőségi jellemző, és olyan belső jellemzők, amelyek mérhetők és összefüggésben állhatnak a külső jellemzőkkel. Az ábra szerint a belső és a külső jellemzők között lehet valamilyen kapcsolat, de nem mondja meg, hogy milyen. Ha a belső jellemző jól használható előrejelzője a külső szoftverjellemzőknek, akkor három feltételnek kell fennállnia (Kitchenham, 1990):

1. A belső jellemzőnek pontosan mérhetőnek kell lennie.
2. Léteznie kell valamilyen kapcsolatnak a mérhető és a bennünket érdeklő külső viselkedésű jellemzők között.
3. A kapcsolat ismert, validált és kifejezhető képlet vagy modell segítségével.

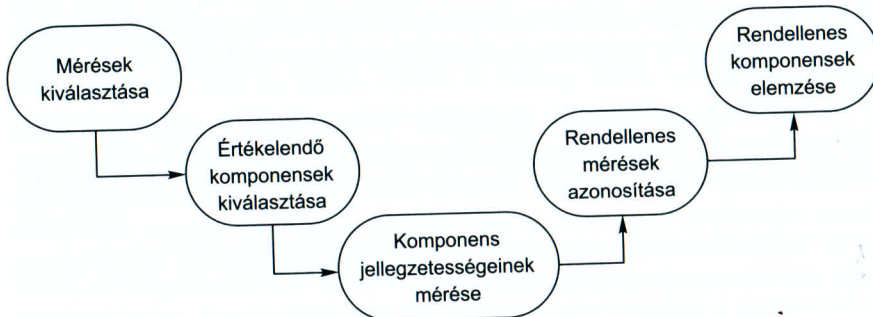
A modell elkészítése magában foglalja az összegyűjtött adatok elemzése révén a modell kiszámítási jellemzőinek (lineáris, exponenciális stb.) meghatározását, a modellben szereplő paraméterek azonosítását, és ezek beállítását a meglévő adatok felhasználásával. Az ilyen típusú megbízható modellek fejlesztésébe ajánlott statisztikus szakembereket is bevonni, mert jelentős tapasztalatra van szükség a statisztikai technikák terén.



27.10. ábra. Összefüggés a belső és külső szoftverjellemzők között

27.5.1. A mérési folyamat

A 27.11. ábrán látható a minőség-ellenőrzési folyamat részét is képezhető szoftver-mérés folyamata. A szoftverek mérésénél a rendszer minden egyes komponensét külön elemzik, és a metrika különböző értékeit összehasonlítják egymással, és lehet, hogy a korábbi projektekből összegyűjtött mérési adatokkal is. A rendellenes mérési eredményeket arra kellene felhasználni, hogy a minőségbiztosítás során a nagyobb figyelmet a minőségileg esetleg problémás komponensekre irányítsák.



27.11. ábra. A termék mérési folyamata

A mérési folyamat fő lépései a következők:

1. *Az alkalmazandó mérések kiválasztása.* Meg kell fogalmazni a kérdést, amelyre a mérés fogja megadni a választ, és meg kell határozni a kérdések megválaszolásához szükséges méréseket. A kérdésekhez nem közvetlenül kapcsolódó mérésekkel nem szükséges foglalkozni. Ennek egy jó megközelítése Basili következő fejezetben tárgyalt átfogó modellje, a GQM (Goal-Question-Metric – Cél-Kérdés-Metrika) (Basili és Rombach, 1988), amely segít annak eldöntésében, milyen adatokat kell összegyűjteni.
2. *A mérni kívánt komponensek kiválasztása.* Nem biztos, hogy egy szoftverrendszer minden komponenséhez metrikus értéket szükséges vagy kívánatos megállapítani. Néhány esetben a komponenseknek csak egy reprezentatív csoportját jelölik ki mérésre, máskor pedig azokat a különösen kritikus komponenseket, amelyeket szinte állandóan használnak (például a központi komponensek).
3. *A komponensek jellemzőinek mérése.* Lemérik a kiválasztott komponenseket, és kiszámítják a metrikus értékeket, ami rendszerint magában foglalja a komponens reprezentációjának (terv, kód stb.) feldolgozását valamilyen automatizált adatgyűjtő eszköz felhasználásával. Ezt az eszközt írhatjuk külön erre a célra, vagy lehet a szervezet által használt CASE-eszközbe beépített eszköz.
4. *A rendellenes mérések azonosítása.* Amint befejeződött a komponensek mérése, a mért értékeket össze lehet hasonlítani egymással és a mérési adatbázisban feljegyzett korábbi mérésekkel. Minden metrikánál meg kell keresni a szokatlanul magas vagy alacsony értékeket, mivel ezek arra figyelmeztetnek, hogy az ilyen értékeket mutató komponensekkel problémák lehetnek.

5. *A rendellenes komponensek elemzése.* A valamely metrikánál rendellenes értéket mutató komponensek feltárása után vizsgálat alá kell vetni őket annak eldöntésére, vajon a rendellenes metrikus értékek veszélyeztetik-e a komponens minőségét. Ha (mondjuk) a bonyolultság rendelkezik ilyen értékkel, akkor ez nem feltétlenül jelenti a komponens gyenge minőségét. Ugyanígy a magas értékből sem szabad a komponens minőségének gyengéségére következtetni, mert a magas értéknek más oka is lehet.

Az összegyűjtött adatokat, mint szervezeti erőforrást, valamint minden projekt történeti jelentését karban kell tartani, még akkor is, ha az adott projekt során nem használták fel ezek adatait. Elég nagy mérési adatbázis felállítása után összehasonlíthatók a projektek, és a specifikus metrikák finomítása segít az olyan információk gyűjtésében, amelyekre szüksége van a minőség növeléséhez.

27.5.2. A termékmetrikák

A termékmetrikák magának a szoftvernek a jellemzőihez kapcsolódnak. Sajnos a könnyen mérhető szoftverjellemzők, mint például a méret vagy a ciklomatikus komplexitás, és a minőségi jellemzők, mint például az érthetőség vagy a karbantarthatóság között nem egyértelmű a kapcsolat és nem általános, hanem a fejlesztési folyamat, a fejlesztési technológia és a fejlesztés alatt álló rendszer típusának függvényében változó. A belső és külső jellemzők közötti kapcsolatok felderítéséhez és validálásához a létező rendszerről sok adatot kell gyűjteni. Ezeket aztán arra lehet használni, hogy kiderítsük, hogyan kapcsolódnak a számunkra érdekes külső minőségek a szoftvertermék jellemzőihez.

A termékmetrikák két osztályba sorolhatók:

1. *Dinamikus metrikák*, amelyeket egy program futása közben készített mérések állítanak elő.
2. *Statikus metrikák*, amelyeket a rendszer reprezentációjának (például terv, program vagy dokumentáció) mérései gyűjtenek össze.

A különböző típusú metrikák különböző minőségi jellemzőkhöz kapcsolódnak. A dinamikus metrikák a program hatékonyságának és megbízhatóságának megállapításában, míg a statikus metrikák a szoftverrendszer komplexitásának, érthetőségének és karbantarthatóságának felmérésében segítenek.

A dinamikus metrikák általában meglehetősen szoros kapcsolatban állnak a szoftver minőségi jellemzőivel. Viszonylag könnyen mérhető az egyes funkciók végrehajtásához és a rendszer elindításához szükséges idő, amelyek közvetlenül is összefüggnek a rendszer hatékonyságával. Hasonlóan a rendszer meghibásodásainak száma és a meghibásodás típusa is naplózható, és a 24. fejezetben leírt módon közvetlen kapcsolatba hozható a szoftver megbízhatóságával.

Ezzel szemben a statikus metrikák és a minőségi jellemzők között csak közvetett kapcsolat létezik. Sok ilyen metrikát ajánlottak már, és kísérleteket is folytat-

tak, hogy megpróbálják megtalálni és rögzíteni a metrikák és a rendszer komplexitása, érthetősége és karbantarthatósága közötti kapcsolatot. A 27.12. ábra több olyan statikus metrikát is bemutat, amelyek a minőségi jellemzők felmérésére szolgálnak. Ezek közül az érthetőséget, a rendszer komplexitását és a karbantart-hatóságot a program vagy a komponens hossza és a vezérlés bonyolultsága jelzi előre a legmegbízhatóbban.

Szoftvermetrika	Leírás
Fan-in/fan-out	A <i>fan-in</i> azon függvények vagy módszerek száma, amelyek valamilyen más függvényt vagy módszert (nevezzük X-nek) hívnak meg. A <i>fan-out</i> az X függvény által meghívott függvények száma. A fan-in magas értéke azt jelenti, hogy az X szorosan kapcsolódik a terv többi részéhez, és megváltoztatása sok helyen ütközéshez vezethet. A fan-out magas értéke jelzi, hogy a meghívott komponensek koordinálásához szükséges vezérlőlogika komplexitása megnövelheti az X általános bonyolultságát.
A kód hossza	A program méretének mértéke. Általában minél nagyobb egy program-komponens kódjának mérete, annál valószínűbb, hogy a komponens bonyolultabb és hibára hajlamos. A kód hossza az egyik legmegbízhatóbb mérték annak előrejelzésére, hogy a komponensben hibák lehetnek.
Ciklomatikus komplexitás	A program vezérlésének bonyolultságát méri, amely összefügghet a program érthetőségével. A ciklomatikus komplexitás kiszámítása a 22. fejezetben található meg.
Azonosítók hossza	A program különböző azonosítóinak átlagos hosszát méri. Minél hosszabb egy azonosító, annál nagyobb az érthetőségének a valószínűsége, ezért a program érthetősége egyenes arányban nő az azonosítók hosszával.
A feltételek egymásba ágyazásának mélysége	Azt méri, hogy a program if utasításai milyen mélyen vannak egymásba ágyazva. A mélyen egymásba ágyazott if utasítások nehezen érthetők, és esetleg hibára hajlamosak is.
Fog index	A dokumentum szavainak és mondatainak átlagos hosszát méri. Minél magasabb a Fog index értéke, annál nehezebb megérteni a dokumentumot.

27.12. ábra. A szoftvertermékek metrikái

A 27.12. ábrán felsorolt metrikák általánosak, de a 27.13. ábrán speciális objektumorientált metrikák láthatók. A legismertebb objektumorientált metrikákat Chidamber és Kemerer (Chidamber és Kemerer, 1994) javasolta, és több eszköz is mérni tudja ezeket. Ezek a metrikák részben a 27.12. ábrán bemutatott régi metrikákból származnak, a többi kizárólag az objektumorientált rendszerek sajátja. El-Amam (El-Amam, 2001) kiváló áttekintést ad az objektumorientált metrikákról, tárgyalja a vonatkozó tanulmányokat, és megállapítja, hogy jelenleg még nem rendelkezünk hatékony eszközökkel annak megértéséhez, hogy az objektumorientált metrikák hogyan kapcsolódnak a külső szoftverminőségekhez.

A speciális metrikák alkalmazhatósága függ a projektól, a minőségkezelési csapat céljaitól és a fejlesztés alatt álló szoftver típusától. Néhány helyzetben a

Objektumorientált metrikák	Leírás
Öröklődési fa mélysége	A különálló szintek száma az öröklődési fában, ahol az alosztályok tulajdonságokat és műveleteket (módszereket) örökölnek a szuperosztálytól. Minél mélyebb az öröklődési fa, annál összetettebb a terv, mivel a fa leveleinél elhelyezkedő objektumosztályok megértéséhez sok különböző objektumosztállyal meg kell ismerkedni.
Módszer fan-in/fan-out	Közvetlenül kapcsolódik a 27.12. ábrán bemutatott fan-in/fan-out-hoz. Lényegében a jelentésük is megegyezik, azonban itt nem árt megkülönböztetni az objektum más módszereiből és a külső módszerekből érkező hívásokat.
Osztályonkénti súlyozott metódusok	Az osztály módszereinek száma, súlyozva az egyes módszerek bonyolultságával. Az egyszerű módszerek bonyolultsága így 1, a nagyobb és összetettebb módszerek bonyolultsága ennél magasabb lesz. Minél magasabb a metrika értéke, annál bonyolultabb az objektumosztály. A bonyolult osztályok nehezebben érthetők, de logikailag sem biztos, hogy összefüggnek, ezért szuperosztályként nem lehet őket ténylegesen újrafelhasználni az öröklődési fában.
Túlterhelt műveletek száma	A szuperosztály azon műveleteinek a száma, amelyeket a szuperosztály valamelyik alosztálya túlterhel. A metrika magas értéke azt jelenti, hogy a használt szuperosztály lehet, hogy nem alkalmas az alosztály szülőjének.

27.13. ábra. Objektumorientált metrikák

27.12. és a 27.13. ábrákon feltüntetett összes metrika használható, de más szituációban esetleg egyik sem felel meg. Amikor a szervezetek a minőségkezelési folyamat részeként szoftvermérést vezetnek be, ki kell tapasztalniuk, mely metrikák felelnek meg legjobban az igényeiknek.

27.5.3. A mérések elemzése

A szoftverrel és a szoftverfolyamattal kapcsolatos mennyiségi adatok gyűjtésekor gyakran felmerül az a probléma, hogy meg kell érteni, mit is jelentenek valójában ezek az adatok. Nagyon könnyű félreértelmezni az adatokat, és könnyű téves következtetésekre jutni. A mérések valódi jelentésének megértéséhez figyelmesen kell elemezni őket.

Az összegyűjtött adatok különbözőféleképpen való értelmezhetőségéhez tekintsük a következő esetet. A könnyebb érthetőség kedvéért folyamatmetrikát használok termékmetrika helyett, de a lényeg ugyanaz; a változtatás lényegét gyakran nehéz megérteni.

Egy vezető, azon feltevésből kiindulva, hogy a változtatáskérelmek, valamint a termék használhatósága és alkalmassága valamilyen módon viszonyul egymáshoz, úgy dönt, megfigyeli a vásárlók változtatáskéréseit. Minél több változtatáskérelem érkezik, annál kevésbé felel meg a szoftver a vásárló igényeinek.

Mivel a változtatáskérelmek feldolgozása és a szoftver megváltoztatása sokba kerül, a szervezet elhatározza, hogy a fogyasztók elégedettségének növelése és ugyanakkor a változtatás költségének csökkentése céljából változtat a folyamaton, amely eredményeképpen jobb termékekre és kevesebb változtatási kérelemre számítanak.

Bevezetik a megváltoztatott folyamatot, amelynek része az is, hogy a vásárlókat nagyobb mértékben vonják be a szoftvertervezési folyamatba. Minden termékre bevezetik a béta-tesztelést, és a vásárlók által igényelt változtatásokat beépítik az átadott termékbe. A terméknek a módosított folyamattal kifejlesztett új változatai is piacra kerülnek. A változtatási kérelmek száma egyes esetekben csökken, más esetekben nő. A vezető megzavarodik, és nem tudja megállapítani, milyen hatással voltak a folyamatban végrehajtott cserék a termék minőségére.

A történetek megértéséhez tudni kell azt is, miért érkeztek változtatási kérelmek. Előfordulhat, hogy a szoftver nem azt csinálja, amit a vásárló vár tőle. De az is lehet, hogy a szoftver nagyon jó, széles körben és gyakran használják, néha még olyan célokra is, amelyre eredetileg nem is tervezték, de mivel sokan használják, természetes, hogy több változtatási kérelem születik.

A harmadik lehetőség szerint a szoftvergyártó szervezet érzékenyen reagál a vásárlók változtatási kérelmeire, ezért a vásárlók elégedettek a nekik nyújtott szolgáltatásokkal. A változtatási kérelmek száma azért olyan magas, mert a vásárlók tudják, hogy a szervezet komolyan veszi a kérelmeket, és hogy javaslataik bekerülhetnek a szoftver későbbi változataiba.

Ha a folyamaton végrehajtott változtatások eredményesek voltak, a szoftvert használhatóbbá és megfelelőbbé tették, ez okozhatja a változtatási kérelmek számának csökkenését. De csökkenheti az is, ha a termék piaci részesedését egy rivális termék szerzi meg, és emiatt kevesebben veszik meg a terméket. A változtatási kérelmek száma nőhet, ha nő a vásárlók száma, és ha a béta-tesztelési folyamat meggyőzte a vásárlókat, hogy a szervezet hajlandó változtatásokat végrehajtani, vagy ha a béta-tesztelés alatt a programot nem a megfelelő módon használták.

A változtatási kérelmek adatainak elemzésekor nem lényeges a változtatási kérelmek számának ismerete, viszont szükséges tudni, kitől jött a kérelem, a vásárló hogyan használja a szoftvert, és hogy miért küldte el a kérelmet. Ismerni kell a befolyásoló külső tényezőket is (például a változtatáskérelmi folyamat módosításai vagy a piac változása). Ezen információk birtokában már ki lehet deríteni, hogy a folyamaton elvégzett változtatások ténylegesen is növelik-e a termék minőségét.

A példából látható, hogy egy termék vagy folyamat mennyiségi adatainak értelmezése bizonytalan, szubjektív folyamat. Az éppen mért folyamatok és termékek nincsenek elszigetelve a környezetüktől, a környezet változása pedig érvénytelenné teheti az adatok összehasonlítását. Az emberi tevékenységekhez kapcsolódó mennyiségi adatoknak nem lehet mindig a tényleges értékét alapul venni, meg kell vizsgálni a mérésekre magyarázatot adó mélyebb okokat is.

Kulcsfogalmak

- A szoftverminőség kezelése annak biztosítására vonatkozik, hogy a szoftver kevés hibát tartalmaz, és megfelel a karbantarthatósági, megbízhatósági, hordozhatósági stb. szabványoknak. Minőségkezelésen a szoftverfolyamatok szabványait előíró minőségbiztosítást, a minőségtervezést, valamint a szoftver és a szabványoknak való megfelelést ellenőrző minőség-ellenőrzést értjük.
- A szervezet minőségi kézikönyvében ajánlott rögzíteni a minőségbiztosítási folyamatok halmazát, ennek alapjául szolgálhat az ISO 9000 szabványok által javasolt általános modell.
- A szoftverszabványok fontosak a minőségbiztosításban, mivel ezek képviselik a „legjobb gyakorlat” egy meghatározását. A minőség-ellenőrzési folyamat hivatott azt ellenőrizni, hogy a szoftverfolyamat és a fejlesztés alatt álló szoftver megfelel-e a szabványoknak.
- A szoftverfolyamat átadásra kerülő részeinek felülvizsgálatát egy csapat végzi, akik ellenőrzik az éppen alkalmazott minőségi szabványok betartását. A felülvizsgálat a minőség megállapításának széles körben használt technikája.
- A szoftvermérés használható a szoftver és a szoftverfolyamat mennyiségi adatainak összegyűjtésére. A szoftvermetrikák összegyűjtött értékeiből következtetések vonhatók le a termék és a folyamatok minőségéről.
- A termékminőségi metrikák különösen akkor értékesek, amikor a minőségi szempontból problémás komponenseket kell kiemelni. Az ilyen komponenseket ajánlott részletesebb elemzésnek alávetni.
- Nincsenek szabványosított vagy általánosan alkalmazható szoftvermetrikák. A szervezeteknek ezek kiválasztását és a mérések eredményének alkalmazását a helyi ismeretek és körülmények függvényében kell elvégezniük.

További irodalom

- Software Quality Assurance: From Theory to Implementation.* Kiváló naprakész áttekintés a minőségbiztosítás elméletéről és gyakorlatáról. Tartalmazza az olyan szabványok tárgyalását, mint az ISO 9001. (Galin, D., 2004, Addison-Wesley.)
- Metrics and Models for Software Quality Engineering, 2nd edition.* Egy átfogó tárgyalását adja a szoftvermetrikák definiálási folyamatának, a termék- és objektumorientált metrikáknak. Tartalmaz olyan matematikai alapvetést is, amely a szoftverméréseken alapuló modellek megértéséhez és fejlesztéséhez szükséges. (Kan, S. H., 2003, Addison-Wesley.)
- „Making sense of measurement for small organisations”. Érdekes cikk a metrikák gyakorlati alkalmazásáról. (Kautz, K., *IEEE Software*, March/April 1999.)
- A Quantitative Approach to Software Management: The ami Handbook.* Kitűnő ismeretető, amely bemutatja, hogyan kell bevezetni egy mérési programot, és hogy ennek eredményei hogyan használhatók fel a folyamat továbbfejlesztéséhez. Sajnos jelenleg új kiadása nincs, csak jobb könyvtárakban hozzáférhető. (Pulford, K. és mások, 1996, Addison-Wesley.)

Feladatok

- 27.1. Magyarázza meg, hogy a magas minőségű szoftverfolyamatok miért eredményeznek magas minőségű szoftvertermékeket. Fejtse ki, hogy ennek a minőségkezelési rendszernek milyen lehetséges problémái lehetnek.
- 27.2. Magyarázza meg, hogyan lehet használni a szabványokat arra, hogy a szervezeti tapasztalatok beépítésével hatékony módszereket nyerjünk a szoftverfejlesztésben. Adja meg az ismereteknek négy olyan típusát, amelyek beépíthetők a szervezeti szabványokba.
- 27.3. Beszéljen a szoftver minőségének értékeléséről a 27.7. ábrán bemutatott minőségi jellemzők alapján! Fejtse ki, hogyan lehetne értékelni az egyes jellemzőket.
- 27.4. Tervezzon meg egy felülvizsgálati észrevételek feljegyzésére használható elektronikus űrlapot, amellyel az észrevételeket elektronikusan lehet elküldeni a felülvizsgálóknak.
- 27.5. Írjon le röviden olyan lehetséges szabványokat, amelyeket a következőkhöz lehetne használni:
 - vezérlőszervezetek használata C-ben, C++-ban vagy Javában;
 - egy egyetemen fél éves projektnek előterjeszhető jelentések;
 - programváltoztatások végrehajtásának és jóváhagyásának folyamata (29. fejezet);
 - új számítógép beszerzésének és installálásának folyamata.
- 27.6. Tegyük fel, hogy ön egy olyan vállalatnál dolgozik, ahol adatbázistermékeket fejlesztenek mikroszámítógépes rendszerekhez, és a szervezet szeretné mennyiségileg kifejezni a szoftverfejlesztést. Írjon egy olyan jelentést, amelyben megfelelő metrikákat ajánl ehhez, és módszert javasol ezek összegyűjtésére.
- 27.7. Magyarázza meg, hogy a tervezési metrikák önmagukban miért nem alkalmasak a tervezési minőség előrejelzésére.
- 27.8. Nézzon utána az irodalomban további javasolt tervezési minőségi metrikáknak a fejezetben megtárgyaltakon kívül. Tanulmányozza őket alaposan, és állapítsa meg, mikor adnak ezek valós értékeket.
- 27.9. Magyarázza meg, miért nehéz validálni a kapcsolatokat az olyan belső termékjellemzők, mint a ciklomatikus komplexitás, és az olyan külső jellemzők között, mint a karbantarthatóság.
- 27.10. Adja meg, hogy az automatizált szoftvermérés hogyan használható az extrém programozás (17. fejezet) támogatására.
- 27.11. Akadályozhatják-e a szoftver szabványok a technológiai innovációt?
- 27.12. Egy munkatársa, aki nagyon jó programozó, olyan szoftvert készít, amelynek nagyon kevés hibája van, de következetesen figyelmen kívül hagyja a szervezet minőségi szabványait. Hogyan kellene a szervezet vezetőinek reagálniuk erre?