

6 SZOFTVERKÖVETELMÉNYEK

TÉMAKÖRÖK

A fejezet célja, hogy bemutassa és különböző szempontok szerint kifejtsa a szoftverrendszerrel szemben támasztott szoftverkövetelményeket. A fejezet elolvasása után:

- érteni fogjuk a felhasználói, illetve rendszerkövetelmények fogalmait és azt, miért kell leírni ezeket a követelményeket különböző módokon;
- meg fogjuk érteni a funkcionális és nemfunkcionális szoftverkövetelmények közötti különbségeket;
- meg fogjuk érteni, hogyan szervezhetők a követelmények az ún. szoftverkövetelmények dokumentumába.

TARTALOM

- 6.1. Funkcionális és nemfunkcionális követelmények
- 6.2. Felhasználói követelmények
- 6.3. Rendszerkövetelmények
- 6.4. Interfész-specifikáció
- 6.5. A szoftverkövetelmények dokumentuma

A rendszer követelményei a rendszer által biztosított szolgáltatásoknak és azok működési megszorításainak leírásai. Ezek a követelmények az ügyfeleknek azokat a rendszerrel szemben támasztott igényeit tükrözik, amelyek segítenek megoldani bizonyos problémákat, mint például egy eszköz vezérlése, sorrend felállítása vagy információ keresése. Ezen szolgáltatások és megszorítások kitalálásának, elemzésének, dokumentálásának és ellenőrzésének folyamatát a *követelmények tervezésének* (RE) nevezzük. Ebben a fejezetben magukra a követelményekre és azok leírásának módjára helyezem a hangsúlyt. A követelménytervezési folyamatot a 4. fejezetben ismertettem, és a 7. fejezetben tárgyalom majd részletesebben.

A *követelmény* elnevezést a szoftveriparban nem használják következetes módon. Néhány esetben a követelmények egyszerűen a rendszer által nyújtandó szolgáltatások magas szintű, absztrakt megfogalmazásai vagy a rendszerrel szemben támasztott követelményei. A másik szélsőséges esetben viszont a rend-

szerfunkciók részletes, formális definíciói. Davis (Davis, 1993) megmagyarázza a különbségek létezésének okát.

Ha egy társaság egy nagyméretű szoftverfejlesztési projektre szeretne szerződést kötni, eléggé absztrakt módon kell meghatározni az igényeit ahhoz, hogy a megoldás ne legyen előre adott. A követelményeket úgy kell meghatározni, hogy több vállalkozó is árajánlatot tehessen a szerződésre, ezzel esetleg többféle módot kínálva az ügyfél igényeinek kielégítésére. Ha a szerződést az egyik vállalkozó elnyerte, meg kell írnia egy részletesebb rendszer-definíciót az ügyfél számára azért, hogy az megértse és jóváhagyhassa, amit a szoftver végezni fog. Mindkét dokumentumot nevezhetjük a rendszer követelménydokumentumának.

A követelmények tervezési folyamata során felmerülő problémák némelyike annak a következménye, hogy nem sikerült tisztán szétválasztani a leírás e különböző szintjeit. Úgy teszek különbséget, hogy a magas szintű absztrakt követelményekre a *felhasználói követelmények* elnevezést használom, a rendszer által végzendő tevékenység részletes leírására pedig a *rendszerkövetelmények* elnevezést. A felhasználói követelmények és rendszerkövetelmények a következőképpen definiálhatók:

1. A *felhasználói követelmények* diagramokkal kiegészített természetes nyelvű kijelentések arról, hogy mely szolgáltatásokat várunk el a rendszertől, és annak mely megszorítások mellett kell működnie.
2. A *rendszerkövetelmények* a rendszer funkcióit, szolgáltatásait és működési megszorításait jelölik ki részletesen. A rendszerkövetelmények dokumentumának (melyet néhol funkcionális specifikációnak is hívnak) pontosnak kell lennie. Pontosán meg kell határozni, mit kell implementálni. Ez a rendszer vásárlója és a szoftverfejlesztő közötti szerződés része lehet.

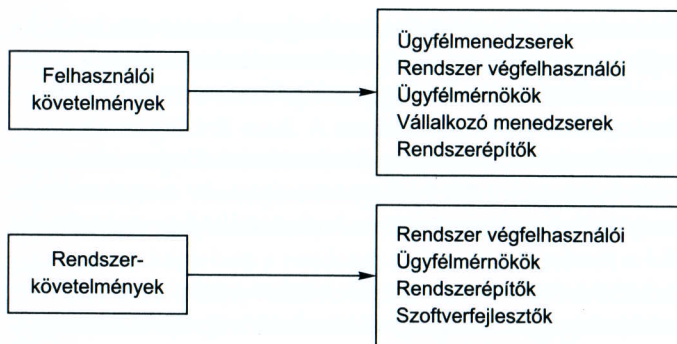
Felhasználói követelmények definíciója

1. A LIBSYS-nek rögzítenie kell minden olyan adatot, amelyre az Egyesült Királyságban és egyéb országokban működő szerzői jogokat felügyelő hivatalok igényt tartanak.

Rendszerkövetelmények specifikációja

- 1.1. Ha dokumentumot kérnek ki a LIBSYS-től, a kérelmező számára biztosítani kell egy űrlapot, amely rögzíti a felhasználó és a kérés részleteit.
- 1.2. A LIBSYS kölcsönzési űrlapokat a kérés időpontjától számított öt évig tárolnia kell a rendszernek.
- 1.3. Az összes LIBSYS kölcsönzési űrlapot indexelni kell a felhasználó, a kért anyag címe és a kérés kiszolgálója szerint.
- 1.4. A LIBSYS-nek naplózni kell a rendszerbe érkező összes kérést.
- 1.5. A szerzői jog hatálya alá eső anyagok kölcsönzéseinek adatait havonta meg kell küldeni a LIBSYS-hez regisztrált szerzői jogokat felügyelő hivataloknak.

6.1. ábra. Felhasználói és rendszerkövetelmények



6.2. ábra. A különböző típusú specifikációk olvasói

A rendszer-specifikáció különböző szintjei azért hasznosak, mert a rendszer-ről különböző típusú olvasók számára közölnek információt. A 6.1. ábra szemlélteti a felhasználói és rendszerkövetelmények közötti eltéréseket. Ez a példa a könyvtári rendszerről megmutatja, hogy a felhasználói követelmények hogyan terjeszthetők ki különféle rendszerkövetelményekké. A 6.1. ábráról leolvasható, hogy a felhasználói követelmények inkább absztraktak, míg a rendszerkövetelmények részletezők, megmagyarázva a fejlesztendő rendszer által biztosítandó szolgáltatásokat és funkciókat.

A követelményeket különböző részletességi szinteken kell megírni, mivel különböző olvasók eltérő módon használják azt. A 6.2. ábra bemutatja a felhasználói és rendszerkövetelmények olvasóinak fajtáit. A felhasználói követelmények olvasói rendszerint nem foglalkoznak a rendszer megvalósításának módjával, és lehetnek olyan vezetők is, akiket nem érdekelnek a rendszer részletes képességei. A rendszerkövetelmények olvasóinak pontosabban kell ismerniük a rendszer működését, mert őket érinti, hogyan támogatja majd a rendszer az üzleti folyamatokat, vagy mert részt vesznek a rendszer implementálásában.

6.1. FUNKCIONÁLIS ÉS NEMFUNKCIONÁLIS KÖVETELMÉNYEK

A szoftverrendszer-követelményeket gyakran felosztják funkcionális és nem-funkcionális követelményekre, illetve szakterületi követelményekre:

1. *Funkcionális követelmények.* Ezek a rendszer által nyújtandó szolgáltatások ismertetései arról, hogyan kell reagálnia a rendszernek bizonyos bemenetekre, és hogyan kell viselkednie a rendszernek bizonyos szituációkban. Néhány esetben a funkcionális követelmények explicit módon azt is megadhatják, mit nem szabad tennie a rendszernek.
2. *Nemfunkcionális követelmények.* Ezek a rendszer által kínált funkciókra és szolgáltatásokra tett megkorlátozások. Magukban foglalnak időbeli korlátozásokat,

a fejlesztési folyamatra tett megkorításokat, szabványokat. A nemfunkcionális követelmények gyakran a rendszer egészére vonatkoznak. Rendszerint nem csupán egyedi rendszerösszetevőkre vagy szolgáltatásokra nézve fogalmaznak meg követelményeket.

3. *Szakterületi követelmények.* Ezek a követelmények a rendszer alkalmazási szakterületéről származnak és e szakterület jellegzetességeit és megkorításait tükrözik. Ezek lehetnek funkcionális vagy nemfunkcionális követelmények.

A valóságban a különböző követelménytípusok közötti különbségtétel nem olyan éles, mint ahogy ezek az egyszerű meghatározások sugallják. Mondjuk, egy biztonságra vonatkozó felhasználói követelmény megjelenhet nemfunkcionális követelményként. Ugyanakkor részletesebben kidolgozva olyan egyéb, tisztán funkcionális követelményekhez vezethet, mint a felhasználói jogosultságokat kezelő eszközök beépítésének szükségessége a rendszerbe.

6.1.1. Funkcionális követelmények

Egy rendszer funkcionális követelményei azt írják le, hogyan kellene működni a rendszernek. Ezek a követelmények a fejlesztett szoftver típusától, a szoftver leendő felhasználóitól és a követelmények lefektetésekor a vállalat által képviselt általános szemlélettől függenek. Ha felhasználói követelményekként vannak kifejezve, rendszerint egészen absztrakt leírások. Ugyanakkor a funkcionális rendszerkövetelmények a rendszerfunkciókat részletesen írják le, kifejtve azok bemeleteit, kimeneteit, kivételeit.

Egy szoftverrendszer funkcionális követelményei számos módon kifejezhetők. Álljon itt példaként a LIBSYS nevű egyetemi könyvtári rendszer néhány funkcionális követelménye. A rendszert egyetemi hallgatók és karok használják más könyvtárakból történő könyv- és dokumentumrendelésre:

1. A felhasználó képes legyen keresni a teljes adatbázisban vagy kiválasztani annak egy részhalmazát.
2. A rendszer nyújtson megfelelő megjelenítőket a felhasználó számára a dokumentumtárban lévő dokumentumok olvasásához.
3. Minden rendeléshez legyen egy egyedi azonosító, amelyet a felhasználó át tudjon másolni saját munkaterületére.

Ezek a funkcionális felhasználói követelmények a rendszer által biztosítandó specifikus eszközöket határoznak meg. Ezek a felhasználói követelmények dokumentumából származnak, és azt mutatják be, hogy a funkcionális követelményeket különböző részletezettséggel is megadhatjuk (ellentétben az 1. és 3. követelménnyel).

A LIBSYS-rendszer egy egységes felület számos cikkadatbázishoz. Lehetővé teszi magazinokban, napilapokban és tudományos folyóiratokban megjelent cikkek másolatának letöltését. A LIBSYS alapját képező rendszer követelményeinek

részletesebb leírása megtekinthető Gerald Kotonyával közösen írt, követelménytervezésről szóló könyvemben (Kotonya és Sommerville, 1998).

A szoftvertervezés számos problémája a követelményspecifikációban lévő pontatlanságokból ered. A rendszerfejlesztő számára természetes, hogy úgy értelmezzen egy kétértelmű követelményt, hogy közben annak megvalósítását egyszerűsítse. Viszont nem biztos, hogy az ügyfél ezt akarta, ezért új követelményeket kell meghatározni, és változtatásokat kell végezni a rendszeren. Természetesen ez késlelteti a rendszer szállítását és növeli a költségeket.

Tekintsük a könyvtári rendszer második követelményét, mégpedig azt, amelyik a rendszer által biztosított „megfelelő megjelenítőre” hivatkozik. A könyvtári rendszer sokféle formátumú dokumentumot kezelhet, e követelménnyel kapcsolatosan pedig az a szándékunk, hogy az összes formátumhoz álljon rendelkezésre megjelenítő. Ugyanakkor a követelmény nem egyértelműen van megfogalmazva; nem teszi világossá, hogy minden egyes dokumentumformátumhoz kell-e megjelenítőt biztosítani. Az ütemterv szorítása alatt dolgozó fejlesztő megteheti, hogy egyszerűen egy szöveges megjelenítőt biztosít és kijelenti, hogy a követelményt teljesítette.

Elvben a rendszer funkcionális követelményt leíró specifikációjának teljesnek és ellentmondásmentesnek kellene lennie. A *teljesség* azt jelenti, hogy a felhasználó által igényelt összes szolgáltatást definiáljuk. Az *ellentmondás-mentesség* azt jelenti, hogy ne legyenek ellentmondó meghatározások. A gyakorlatban nagyméretű, összetett rendszereknél gyakorlatilag lehetetlen a követelmények teljességét és ellentmondás-mentességét elérni.

Ennek egyik oka, hogy nagyméretű, összetett rendszerek specifikációinak írásakor könnyű kifelejtetni dolgokat, illetve hibát elkövetni. Másik ok, hogy a különböző kulcsfiguráknak (7. fejezet) eltérő – és gyakran ellentmondó – igényeik vannak. Elképzelhető, hogy ezek az ellentmondások nem derülnek ki azonnal, így az ellentmondásos követelmények a specifikációban maradnak. A problémák csak mélyebb elemzés után jelentkeznek, vagy néha csak a fejlesztés befejezte és a rendszer átadása után.

6.1.2. Nemfunkcionális követelmények

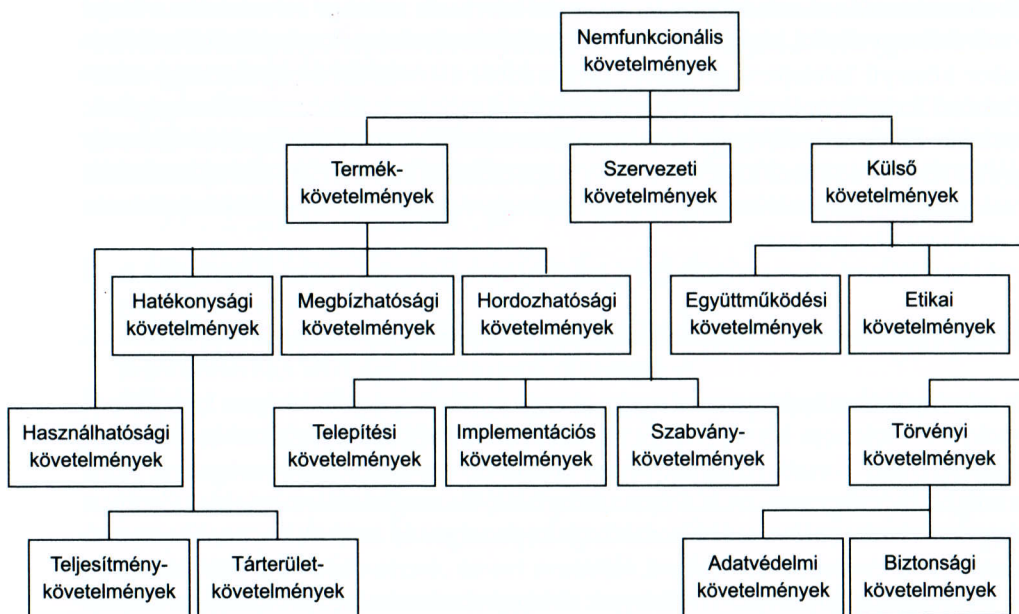
A nemfunkcionális követelmények, ahogy a név is sugallja, olyan követelmények, amelyek nem közvetlenül a rendszer által szállított specifikus funkciókkal foglalkoznak. Vonatkozhatnak olyan eredendő rendszertulajdonságokra, mint a megbízhatóság, a válaszidő és a tárfoglalás, de meghatározhatnak akár olyan megszorításokat, mint az I/O-eszközök képességei és a rendszerinterfészekben használt adatrepresentációk.

A nemfunkcionális követelmények ritkán vonatkoznak egyedi rendszerösszetevőkre. Ezek a követelmények inkább a rendszer eredendő tulajdonságait írják le, vagy azokra adnak megszorítást, ahogyan azt a 2. fejezetben tárgyaltam. Ezért ezek leírhatják a rendszer teljesítményét, biztonságosságát, rendelkezésre állását és egyéb eredendő tulajdonságot. Ez azt jelenti, hogy ezek gyakran kritikusabbak,

mint az egyedi funkcionális követelmények. A rendszer felhasználói rendszerint megtalálják a módját, hogyan kerüljek meg azt a rendszerfunkciót, amely nem igazán megfelelő nekik. Ugyanakkor egy nemfunkcionális rendszerkövetelmény teljesítésének kudarca a teljes rendszert használhatatlanná teheti. Például ha egy repülőgéprendszer nem teljesíti a vele szemben támasztott megbízhatósági követelményeket, nem kapja meg a biztonsági tanúsítványt a működéséhez; ha egy valós idejű vezérlőrendszer nem elégíti ki a teljesítményével szemben támasztott követelményeket, a vezérlési funkciók nem fognak helyesen működni.

A nemfunkcionális követelmények nem csupán a fejlesztendő szoftverrendszerrel foglalkoznak. Egyes nemfunkcionális követelmények behatárolhatják azt a folyamatot, amit a rendszer fejlesztésekor kell használnunk. A folyamat követelményeire vonatkozó példák magukban foglalhatják a folyamat során használandó minőségsszabványokat, előírhatják, hogy a tervet egy bizonyos CASE-eszközzel kell elkészíteni, és tartalmazhatják a követendő folyamat leírását.

Nemfunkcionális követelményeket felvethetnek felhasználói igények, költségvetési megkorlátozások, a szervezeti szabályzat, más szoftver- vagy hardverrendszerekkel való együttműködés igénye vagy olyan külső tényezők, mint a biztonsági szabályozások, adatvédelmi rendelkezések. A 6.3. ábrán a nemfunkcionális követelmények osztályozása látható. Az ábrán látszik, hogy a nemfunkcionális követelmények eredhetnek a szoftver kívánt jellemzőiből (termékkövetelmények), a szoftvert fejlesztő vállalattól (vállalati követelmények) vagy külső forrásokból.



6.3. ábra. Nemfunkcionális követelmények típusai

A nemfunkcionális követelmények típusai:

1. *Termékre vonatkozó követelmények.* Ezek a követelmények határozzák meg a termék viselkedését. A példákban szerepelnek teljesítményre vonatkozó követelmények arról, milyen gyorsan kell futnia és mennyi memóriára van szüksége a rendszernek; megbízhatósági követelmények, amik megszabják az elfogadható meghibásodási arányt; hordozhatósági követelmények és használhatósági követelmények.
2. *Szervezeti követelmények.* Ezek a követelmények a megrendelő és a fejlesztő szervezetének szabályzataiból és ügyrendjéből erednek. A példák tartalmazzanak felhasználandó folyamatszabványokat; megvalósítási követelményeket, mint például a használt programozási nyelv vagy tervezési módszer; és szállítási követelményeket, amelyek megadják a termék és dokumentációja átadási idejét.
3. *Külső követelmények.* Ez tág címszó, minden olyan követelmény idetartozik, amely a rendszeren és annak fejlesztési folyamatán kívüli tényezőtől származik. Ezek magukban foglalhatnak együttműködési követelményeket, amelyek meghatározzák, hogyan érintkezik a rendszer más szervezetek rendszereivel; törvényi követelményeket, amelyeket követni kell a rendszer jogi keretek közötti működésének biztosításához; továbbá etikai követelményeket. Az etikai követelmények olyan, a rendszerrel szemben támasztott követelmények, amelyek biztosítják a rendszernek a felhasználók és a közvélemény számára elfogadható működését.

A 6.4. ábra termék-, szervezeti és külső követelményekre mutat példát. Ezek annak a LIBSYS könyvtári rendszernek a követelményei, amelynek a felhasználói követelményeit a 6.1.1. alfejezetben tárgyaltam. A termékkövetelmény korlátozza a LIBSYS tervezőinek szabadságát a rendszer felhasználói felületének megvalósításában. Semmit nem mond a LIBSYS funkcióiról, és jól láthatóan egy rendszermegszorítást azonosít, nem pedig egy funkciót. A követelmény azért lett megszabva, mert így egyszerűbben biztosítható, hogy a rendszer különböző bön-
gészőkkel is működjék.

Termékkövetelmény

8.1. A LIBSYS felhasználói felületét egyszerű HTML formában kell megvalósítani, keretek és Java-appletek nélkül.

Szervezeti követelmény

9.3.2. A rendszerfejlesztési folyamat és az átadandó dokumentumok feleljenek meg az XYZCo-SP-STAN-95-ben definiált folyamatnak és átadandó dokumentumoknak.

Külső követelmény

10.6. A rendszer ne fedjen föl a könyvtári dolgozók előtt semmilyen személyes információt a rendszer felhasználóiról a nevükön és a könyvtári azonosító számukon kívül.

6.4. ábra. Példák nemfunkcionális követelményekre

A szervezeti követelmény arról szól, hogy a rendszert az XYZCo-SP-STAN-95 vállalati szabványnak megfelelően kell fejleszteni. A külső követelményt az az igény hozta létre, hogy a rendszer megfeleljen az adatvédelmi törvényeknek. Ez kimondja, hogy nem szabad a rendszer operátorai számára hozzáférhetővé tenni olyan adatokat (például a rendszer felhasználóinak lakcímét), amelyekre a munkájuk során nincs szükségük.

A nemfunkcionális követelményekkel kapcsolatos általános probléma, hogy esetenként nehéz lehet verifikálni őket. A felhasználók és a megrendelők ezeket a követelményeket gyakran olyan általános célokként fogalmazzák meg, mint a könnyű használat, a rendszer összeomlás utáni visszaállítási képessége vagy a gyors válaszidő. Ezek a követelmények gondot okoznak a rendszerfejlesztőknek, mivel teret hagynak az eltérő értelmezéseknek és a későbbi vitáknak a rendszer átadása után. E probléma szemléltetésére tekintsük a 6.5. ábrát. Ez egy forgalomirányító rendszer használhatóságára vonatkozó rendszercélt mutat be, amely jellemző arra, ahogyan a felhasználók használhatósági követelményeket szoktak megfogalmazni. Ezt úgy írtam át, hogy látszódjék, hogyan fejezhető ki egy cél „tesztelhető” nemfunkcionális követelményként. Amíg a rendszercélt objektíven verifikálni lehetetlen, addig lehet olyan rendszerteszteket tervezni, amelyek egy szimulátor segítségével számolják a vezérlő által elkövetett hibákat.

Amikor csak lehet, igyekezzünk a nemfunkcionális követelményeket mennyiségiel, objektíven tesztelhető metrika segítségével kifejezni. A 6.6. ábra néhány lehetséges metrikát mutat be, amiket a nemfunkcionális rendszertulajdonságok megadására használhatunk. A rendszer tesztelése alatt ezeket a jellemzőket mérve megállapíthatjuk, hogy a rendszer kielégíti-e a nemfunkcionális követelményeit vagy sem.

A valós életben viszont a rendszer megrendelői számára gyakorlatilag lehetetlen céljaikat mennyiségi követelményekre váltani. Néhány célhoz, mint például a karbantarthatóság, nincs használható metrika. Máskor, még ha a mennyiségi specifikáció lehetséges is, elképzelhető, hogy a megrendelők nem tudják igényeiket ezekkel a specifikációkkal kifejezni. Nem értik, hogy egy bizonyos, mondjuk a kívánt megbízhatóságot meghatározó szám mit jelent majd az ő mindennapi, számítógépes rendszerekkel végzett munkájuk során. Továbbá a mennyiségi nemfunkcionális követelmények objektív verifikálásának költsége nagyon magas lehet, és a rendszert fizető megrendelők azt gondolhatják, hogy ez a költség nem fog megtérülni.

Rendszercél

A rendszernek a tapasztalt ellenőrök számára könnyen használhatónak kell lennie, és javallott olyan módon szervezni azt, hogy a felhasználói hibák száma a lehető legkisebb legyen.

Verifikálható nemfunkcionális követelmény

A tapasztalt ellenőrök képesek legyenek az összes rendszerfunkció használatára egy kétórás képzés után. A képzés után a tapasztalt felhasználók által elkövetett hibák száma átlagosan ne haladja meg a napi kettőt.

6.5. ábra. Rendszercélok és verifikálható követelmények

Tulajdonság	Mérés
Sebesség	Feldolgozott tranzakciók/másodperc Felhasználó/esemény válaszidő Képernyő-frissítési idő
Méret	Kbyte RAM-chipek száma
Könnyű használhatóság	Képzési idő Help képernyők száma
Megbízhatóság	Hibák átlagos bekövetkezési ideje A rendelkezésre nem állás valószínűsége Hibák bekövetkeztenek aránya Rendelkezésre állás
Robusztusság	A hiba utáni újraindulási idő A hibát okozó események százalékos aránya A hiba okozta adatmeghibásodás valószínűsége
Hordozhatóság	A célrendszerfüggetlen utasítások százalékos aránya A célrendszerek száma

6.6. ábra. Nemfunkcionális követelmények megadásához használható metrikák

Ezért a követelménydokumentumok gyakran a célokat a követelményekkel vegyesen tartalmazzák. Ezek a célok hasznosak lehetnek a fejlesztők számára, mert vezérfonalat nyújtanak ahhoz, hogy a megrendelő mit lát fontosnak. Ugyanakkor közölni kell az ügyféllel, hogy ezek félreértelmezésekre adhatnak alapot és objektíven nem ellenőrizhetők.

A nemfunkcionális követelmények gyakran ütköznek más funkcionális vagy nemfunkcionális követelményekkel, vagy hatnak azokra. Például követelmény lehet, hogy a rendszer által elfoglalt tár legfeljebb 4 MB lehet. A memóriakövetelmények megszokottak a beágyazott rendszereknél, ahol korlátozva van a méret vagy a súly, ezért a rendszerszoftvert tároló ROM-lapok számát minimalizálni kell. További követelmény lehet, hogy a rendszer Adában, egy kritikus valós idejű szoftverfejlesztéshez tervezett programozási nyelven íródjon. Ugyanakkor nem lehetséges egy, a megkövetelt funkciókat teljesítő Ada-programot 4 MB-nál kisebbre lefordítani. Ezenél a követelményeknél valamilyen kompromisszumot kell kötni: vagy másik programnyelvet kell használni, vagy további memóriával kell bővíteni a rendszert.

A funkcionális és nemfunkcionális követelményeket célszerű megkülönböztetni a követelménydokumentumban. A gyakorlatban ezt bonyolult megtenni. Ha a nemfunkcionális követelmények a funkcionális követelményektől külön helyezkednek el, néha nehezen láthatók a köztük lévő összefüggések. Ha a funkcionális követelményekkel együtt állnak, nehéz lehet elkülöníteni a funkcionális és nemfunkcionális megfontolásokat és azonosítani a rendszer egészére vonatkozó követelményeket. Ugyanakkor, a tisztán az eredendő rendszertulajdonságokra (például teljesítmény vagy megbízhatóság) vonatkozó követelményeket javallott

explicit módon kiemelni. Vagy úgy, hogy a követelménydokumentum külön részében helyezzük el őket, vagy úgy, hogy valahogyan megkülönböztetjük őket az egyéb rendszerkövetelményektől.

A nemfunkcionális követelmények, mint a védetség és a biztonságossági követelmények, különösen fontosak a kritikus rendszereknél. Ezért a megbízhatósági követelményeket részletesebben tárgyalom majd a kritikus rendszerek specifikációjával foglalkozó 9. fejezetben.

6.1.3. Szakterületi követelmények

A szakterületi követelmények inkább a rendszer alkalmazásának szakterületéből származnak, nem pedig a rendszer felhasználóinak egyéni igényeiből. Ezek rendszerint speciális szakterületi terminológiát vagy a szakterület fogalmaira történő utalásokat tartalmaznak. Ezek lehetnek új, teljes jogú funkcionális követelmények, megszorítást írhatnak elő létező funkcionális követelményeken, vagy kijelölhetik, hogyan kell bizonyos számításokat véghezvinni. Mivel ezek speciális követelmények, a szoftvertervezőknek sokszor nehéz megérteniük, hogyan viszonyulnak ezek más rendszerkövetelményekhez.

A szakterületi követelmények jelentősége nagy, mivel gyakran az alkalmazási szakterület alapjait tükrözik. Ha ezek a követelmények nem teljesülnek, nem lehetséges a rendszert kielégítően működtetni. A LIBSYS-rendszer számos szakterületi követelményt tartalmaz:

1. Minden adatbázishoz legyen szabványos felhasználói felület, amelynek a Z39.50 szabványon kell alapulnia.
2. Szerzői jogi korlátozások miatt néhány dokumentumot a megérkezése után azonnal törölni kell. A felhasználói követelményektől függően ezek a dokumentumok vagy helyben, a rendszer szerverén kerülnek kinyomtatásra, hogy aztán kézzel továbbítsák azokat az olvasóhoz, vagy egy hálózati nyomtatóra irányítják át.

Az első követelmény egy tervezési megszorítás. Ez megadja, hogy a felhasználói felületet az adatbázishoz egy adott könyvtári szabvány szerint kell megvalósítani. A fejlesztőknek ezért előbb meg kell ismerkedniük a szabvánnyal, mielőtt a felület megtervezéséhez hozzáfognának. A második követelményt szerzői jogi törvények miatt vezették be, melyek a könyvtárakban használt anyagokra vonatkoznak. Eszerint a könyvtári rendszernek automatikus „nyomtatás-után-töröld” eszközt kell tartalmaznia a dokumentumok bizonyos osztályai számára. Ez azt jelenti, hogy a könyvtári tagok nem kaphatnak elektronikus másolatot a dokumentumról.

A számítás véghezvitelének módját megadó szakterületi követelmények szemléltetéséhez tekintsük a 6.7. ábrát, amely egy automatizált vasúti védelmi rendszer specifikációjából származik. Ez a rendszer önműködően megállítja a vonatot, ha az vörös jelzésen halad át. Ez a követelmény azt fogalmazza meg, ho-

A vonat lassulása a következőképpen számítandó:

$$D_{\text{vonat}} = D_{\text{vezérlő}} + D_{\text{gradiens}}$$

ahol a D_{gradiens} $9,81 \text{ ms}^{-2} \cdot \text{kompenzált gradiens/alfa}$, ahol a $9,81 \text{ ms}^{-2}/\text{alfa}$ -értékek a különböző típusú vonatok esetén ismertek.

6.7. ábra. Egy vasúti védelmi rendszerről szóló szakterületi követelmény

gyan számítja ki a rendszer a vonat lassulását. A szakterület a saját terminológiáját használja. A megértéséhez szükségünk van a vasúti rendszerek működésének és a vonatok jellemzőinek némi ismeretére.

A vasúti rendszer követelménye szemlélteti a szakterületi követelményekkel kapcsolatos legfőbb problémát. A követelményeket az alkalmazási szakterület nyelvén írták le (ez esetben matematikai egyenletekkel), és a szoftvertervezők számára gyakran nehéz megérteni őket. A szakterület szakértői gyakran kihagyhatnak információkat a követelményből, mivel az számukra teljesen nyilvánvaló. Ugyanakkor a rendszer fejlesztői számára nem nyilvánvaló, és előfordulhat, hogy ezért a követelménynek rosszul tesznek eleget.

6.2. FELHASZNÁLÓI KÖVETELMÉNYEK

A rendszer felhasználói követelményeinek a funkcionális és nemfunkcionális követelményeket kell leírniuk úgy, hogy a rendszer azon felhasználói is megértsék azokat, akiknek nincsenek részletes technikai ismereteik. A rendszernek csak a külső viselkedését javallott leírniuk, és amennyire lehet, kerülni kell a rendszer tervezésének jellemzőit. Ezért a felhasználói követelmények írásakor nem célszerű számítógépes zsargont, strukturált jelöléseket vagy formális jelöléseket használni, illetve a követelményt a rendszer megvalósításának leírásával megadni. A felhasználói követelményeket egyszerű nyelven, egyszerű táblázatok, űrlapok és könnyen érthető diagramok segítségével kell közreadni.

Ugyanakkor a természetes nyelven írt követelményeknél változatos problémák merülhetnek föl:

1. *Az egyértelműség hiánya.* Olykor nehéz a nyelvet pontos, egyértelmű módon használni anélkül, hogy a dokumentumot terjengőssé és nehezen olvashatóvá ne tennénk.
2. *Követelmények keveredése.* A funkcionális követelmények, nemfunkcionális követelmények, rendszercélok és tervezési információk nem különíthetők el tisztán.
3. *Követelmények ötvöződése.* Több különböző követelmény egyetlen követelményként fogalmazódik meg.

4.5. A könyvtári rendszernek biztosítania kell egy pénzügyi számlázó rendszert, amely nyilvántartja a rendszer felhasználóinak kifizetéseit. A rendszer kezelői úgy konfigurálhatják a rendszert, hogy a normál felhasználók árengedményt kaphassanak.

6.8. ábra. A LIBSYS számlázási rendszerével szembeni felhasználói követelmény

Ezek közül néhány probléma szemléltetéséhez tekintsük a 6.8. ábrán szereplő könyvtári követelmények egyikét.

Ez a követelmény fogalmi és részletezett információkat is magában foglal. Azt a koncepciót fejezi ki, hogy a LIBSYS szerves részét kell képezze egy számlázó-rendszer. Ugyanakkor azt a részletet is magában foglalja, hogy a számlázórendszernek támogatnia kell árengedmények adását bizonyos LIBSYS-felhasználóknak. Ezt a részletet jobb lett volna a rendszerkövetelmény-specifikációra hagyni.

Jó gyakorlat, hogy egy követelménydokumentumon belül a felhasználói követelményeket elkülönítjük a részletesebb rendszerkövetelményektől. Másikülönben a felhasználói követelmények nem szakértő olvasóit eláraszthatják a valójában csak a szakemberek számára érdekes részletek. A 6.9. ábra szemlélteti ezt a keveredést. Ez a példa szoftvertervmodellek szerkesztésére szolgáló CASE-eszköz követelménydokumentumból származik. A felhasználó megadhatja, hogy megjelenjen egy rács azért, hogy az egyedeket a diagramon pontosan el lehessen helyezni.

2.6. Rács eszközök Az egyedek diagramon történő pozicionálását segítő a felhasználó bekapcsolhat egy rácsot akár centiméteres, akár hüvelykes beosztással, a vezérlőpanelen lévő opció segítségével. Kezdetben a rács kikapcsolt állapotban van. A rács a szerkesztés menete alatt tetszőlegesen ki-be kapcsolható, és bármikor válthatunk a centiméteres és hüvelykes beosztás között. A rács opció az ablakméretre illeszkedő nézetben is elérhető lesz, de a rácsvonalak száma csökkenni fog annak elkerülésére, hogy a kisebb diagramokat kitöltsék a rácsvonalak.

6.9. ábra. Egy szerkesztőrácsra vonatkozó felhasználói követelmény

Az első mondat három különböző követelményt mos egybe.

1. Egy fogalmi, funkcionális követelményt, amely kijelenti, hogy a szerkesztő-rendszernek rácsot kell biztosítania. Erre magyarázatot is ad.
2. Egy nemfunkcionális követelményt, mely részletes információt ad a rács egységeiről (centiméter vagy hüvelyk).
3. Egy nemfunkcionális, felhasználói felületi követelményt, amely meghatározza, hogyan kapcsolhatja a rácsot ki és be a felhasználó.

A 6.9. ábrán szereplő követelmény tehát ad néhány információt a kezdő beállításról, de nem adja meg valamennyit. Meghatározza, hogy a rács kezdetben ki van kapcsolva. Ugyanakkor, nem határozza meg a bekapcsoláskor érvényes

egységet. Nyújt némi részletezett információt, nevezetesen, hogy a felhasználó válthat az egységek között, de a rácsvonalak közötti távolságot nem rögzíti.

Ha a felhasználói követelmények túl sok információt tartalmaznak, az korlátozza a rendszerfejlesztő szabadságát abban, hogy újító megoldással szolgáljon a felhasználói problémákra, és nehezen érthetővé teszi a követelményeket. A felhasználói követelményeknek egyszerűen a kulcsfontosságú igényekre kell összpontosítani. Ezt szemlélteti a 6.10. ábra, melyben a szerkesztőrácsra vonatkozó követelményt átirтам úgy, hogy az csak a lényeges rendszertulajdonságokra helyezze a hangsúlyt.

2.6.1. Rács eszközök

A szerkesztő biztosítson egy rács eszközt, ahol a vízszintes és függőleges vonalak mátrixa háttérrel nyújt a szerkesztőablakhoz. Ez a rács legyen passzív rács, ahol az elemek igazítása a felhasználóra tartozik.

Magyarázat: Egy rács segíti a felhasználót egy rendezett diagram elkészítésében, jól elhelyezett elemekkel. Bár egy aktív rács, ahol az elemek a rácsvonalakhoz „tapadnak”, hasznos lehet, de a pozicionálást pontatlanná teszi. Annak eldöntésében, hova kerüljenek az elemek, a felhasználó a legilletékesebb személy.

Specifikáció: ECLIPSE/WS/Tools/DE/FS 5.6. szakasz

Forrás: Ray Wilson, Glasgow Office

6.10. ábra. Egy szerkesztőrács eszköz definíciója

Hacsak lehet, minden felhasználói követelményhez kapcsoljunk magyarázatot. Ez a magyarázat megmondja, miért került be a követelmény, és különösen hasznos, amikor a követelmények megváltoznak. Például a 6.10. ábrán a magyarázat megmutatja, hogy hasznos lehet egy aktív rács, ahol a pozicionált objektumok azonnal a rácsvonalhoz tapadnak. Ugyanakkor tudatosan elveti ezt a lehetőséget, és a kézi pozicionálás mellett dönt. Ha ezen valamelyik későbbi szakaszban változtatni szándékozunk, akkor világos, hogy a passzív rács használata tudatos volt, nem pedig egy implementációs döntés.

A felhasználói követelmények írásakor felmerülő félreértések száma csökkenthető, ha betartunk néhány egyszerű irányelvet:

1. Találjunk ki egy szabványos formátumot, és az összes követelménydefiníció igazodjon ehhez a formátumhoz. A formátum szabványosításának az a jelentősége, hogy kevésbé valószínűek lesznek az elhagyások, és a követelmények könnyebben ellenőrizhetővé válnak. Az általam használt formátum magában foglalja a kezdő követelmény félkövér kiemelését, beleértve a magyarázat megfogalmazását is minden felhasználói követelményhez és egy hivatkozást a rendszerkövetelmények részletesebb specifikációjához. Arról is tehetünk bele információt, hogy ki javasolta a követelményt (a követelmény forrása), azért, hogy a követelmény változása esetén tudjuk, kivel kell konzultálni.

2. Használjuk következetesen a nyelvet. Tegyük különbséget a szükséges és kívánatos követelmények között. A *szükséges* követelmények azok, amelyeket a rendszernek támogatnia kell, ezeket a „kell” szóval szokták kifejezni. A *kívánatos* követelmények nem döntő fontosságúak, és a „javallott” szóval szokás kifejezni őket.
3. Használjunk szövegkiemelést (félkövér, dőlt betűk vagy színezés) a követelmények kulcsfontosságú részeinek hangsúlyozására.
4. Kerüljük el, amennyire lehet, a számítógépes zsargon használatát. Ugyanakkor elkerülhetetlen, hogy részletes szakkifejezések is bekerüljenek a felhasználói követelményekbe.

Robertsonék (Robertson és Robertson, 1999) a VOLERE követelménytervezési módszert tárgyaló könyvükben azt javasolják, hogy a felhasználói követelményeket először kártyára írjuk, minden lapra egyet. Minden lapra javasolnak felvenni néhány mezőt, például a követelmény magyarázatát, más követelményektől való függést, a követelmény forrását, támogatási anyagokat és így tovább. Ez kibővíti a 6.10. ábrán használt formát, és felhasználói és rendszerkövetelményekhez egyaránt használható.

6.3. RENDSZERKÖVETELMÉNYEK

A rendszerkövetelmények a felhasználói követelmények kibővített változatai, amelyeket a szoftvertervezők a rendszerterv kiindulási pontjaként használnak. Megmagyarázzák, hogy a rendszernek hogyan kell biztosítani a felhasználói követelményeket. Alapul szolgálnak a rendszer megvalósítási szerződéséhez, és ezért az egész rendszer teljes és konzisztens meghatározását tartalmazniuk kell.

A rendszerkövetelmények ideális esetben egyszerűen a rendszer külső viselkedését és működési megkorlátait írják le. A tervezés és az implementálás mikéntjével nem szabad foglalkozniuk. Ennek ellenére a részletességnek azon a szintjén, amely egy összetett szoftverrendszer teljes meghatározásához szükséges, lehetetlen kizárni minden tervezési információt. Ennek számos oka van:

1. A követelményspecifikáció strukturálását segítő szükségünk lehet a rendszer egy kiindulási architektúrájának megtervezésére. A rendszerkövetelmények a rendszert felépítő különböző alrendszereknek megfelelően vannak szervezve. Ahogyan azt a 7. és 18. fejezetben is tárgyalom, az architekturális definíció alapvető fontosságú, ha a rendszer implementálásához újra fel akarunk használni szoftverkomponenseket.
2. A legtöbb esetben a rendszernek együtt kell működnie más, meglévő rendszerekkel. Ezek korlátozhatják a tervezést, és ezek a korlátok követelményeket hoznak létre az új rendszerhez.

3. A nemfunkcionális követelmények kielégítéséhez szükség lehet egy specifikus architektúra használatára (mint az N verziós programozás a megbízhatóság eléréséhez, amit a 20. fejezetben tárgyalok). Egy külső ellenőr, akinek a rendszer biztonságosságát kell bevizsgálnia, előírhatja egy már bevizsgált architektúrális terv használatát.

A természetes nyelvet gyakran használják rendszerkövetelmények specifikációjának és felhasználói követelményeknek a megírásához egyaránt. Ugyanakkor, mivel a rendszerkövetelmények a felhasználói követelményeknél részletesebbek, a természetes nyelvi specifikációk félrevezetőek és nehezen érthetőek lehetnek:

1. A természetes nyelv megértése azon alapszik, hogy a specifikáció olvasói és írói ugyanazokat a szavakat használják ugyanazon fogalmakhoz. Ez félreértésekhez vezet a természetes nyelv többértelműsége miatt. Jackson (Jackson, 1995) kitűnő példát ad erre, amikor mozgólépcső mellett látott reklámokat és feliratokat tárgyal. Ezek az állt: „Használt cipőt vegyenek fel!” és „A kutyát vegyék fel!”. A két mondat ellentétes értelmezésének megfejtését az olvasóra hagyom.
2. A természetes nyelvű követelményspecifikáció túl rugalmas. Ugyanazt a dolgot teljesen különbözőféleképpen elmondhatjuk. Az olvasó dolga, hogy kitalálja, hogy a követelmények mikor egyeznek meg és mikor különbözőek.
3. A természetes nyelvű követelmények modularizálására nincs könnyű módszer. Lehet, hogy bonyolult az összes kapcsolódó követelményt megtalálni. Ahhoz, hogy felfedezzük egy változtatás következményét, előfordulhat, hogy minden követelményt meg kell néznünk, nem pedig csak az ahhoz kapcsolódó követelmények egy csoportját.

E problémák miatt a természetes nyelven írt követelményspecifikációk félreértésekre adhatnak okot. Ezek gyakran csak a fejlesztési folyamat későbbi fázisaiban derülnek ki, és akkor már igen költséges lehet a feloldásuk.

Döntő fontosságú, hogy felhasználói követelményeket a nem szakemberek számára is érthető nyelven fogalmazzuk meg. A rendszerkövetelményeket ugyanakkor speciális jelölésekkel is kifejezhetjük (6.11. ábra). Ezek közé tartozik a stilizált, strukturált természetes nyelv és a követelmények grafikus modelljei, mint például a használati esetek a formális matematikai specifikációkhoz. Ebben a fejezetben megtárgyalom, hogy az egyszerű grafikus modellekkel kiegészített természetes nyelv hogyan használható rendszerkövetelmények írására. A grafikus rendszermodellezést a 8. fejezetben, a formális rendszer-specifikációt pedig a 10. fejezetben tárgyalom.

Jelölés	Leírás
Strukturált természetes nyelv	Ez a megközelítés függ attól, hogy milyen szabványos űrlapokat vagy sablonokat használunk a követelményspecifikáció kifejezésére.
Tervleíró nyelvek	Ez a megközelítés egy programozási nyelvre hasonlító nyelvet használ, de absztraktabb elemekkel, amik a követelményeket a rendszer egy működési modelljének megadásával írják le. Ez a megközelítés ma nem elterjedt, bár interfész specifikációjához hasznos lehet.
Grafikus jelölések	Egy szöveges magyarázatokkal kiegészített grafikus nyelv, amit a rendszer funkcionális követelményeinek definiálására használnak. Ilyen grafikus nyelvre korai példa az SADT (Ross, 1997; Schoman és Ross, 1977). Újabban a használati eset leírásokat (Jacobsen és mások, 1993) és a szekvenciadiagramokat használják jellemzően (Stevens és Pooley, 1999).
Matematikai specifikációk	Vannak jelölések, melyek olyan matematikai fogalmakon alapulnak, mint a véges állapotú automaták vagy a halmazok. Ezek az egyértelmű specifikációk csökkentik a megrendelő és a vállalkozó közötti vitákat a rendszer funkciójáról. Ugyanakkor a legtöbb megrendelő nem érti a formális specifikációt, és vonakodik elfogadni azt rendszerszerződként.

6.11. ábra. A követelményspecifikációban használható jelölések

6.3.1. Strukturált nyelvű specifikáció

A strukturált természetes nyelv a rendszerkövetelmények leírásának egy módja, ahol a követelmény írójának korlátozott a szabadsága, és minden követelmény valamilyen szabványos módon van leírva. Ennek a megközelítésnek az előnye, hogy a természetes nyelv kifejezőképességének és érthetőségének legnagyobb részét megtartja, de nyújt bizonyos fokú egységességet is a specifikációra épülve. A strukturált nyelvi jelölések korlátot szabnak a használt terminológiának, és sablonokat használhatnak a rendszerkövetelmények megadására. Megtestesíthetnek programozási nyelvekből származó vezérlési szerkezeteket és képi kiemelést a specifikáció részekre bontására.

Egy, a rendszerkövetelmények megadására strukturált természetes nyelvet használó projektet írt le Heninger (Heninger, 1980). Speciális célú űrlapokat terveztek egy repülőgépes szoftverrendszer bemenetének, kimenetének és funkcióinak leírására. A rendszerkövetelményeket az űrlapok segítségével adták meg.

Ahhoz, hogy az űrlalapalapú megközelítést használjuk a rendszerkövetelmények megadásához, definiálnunk kell egy vagy több szabványos űrlapot vagy sablont a követelmények kifejezéséhez. A specifikációt építhetjük a rendszer által manipulált objektumok köré, a rendszer által végzett funkciók köré vagy a rendszer által feldolgozott események köré. Egy ilyen űrlalapalapú specifikációra a 6.12. ábrán látható példa. Ezt a példát a 3. fejezetben bevezetett inzulinadagoló rendszerből vettem.

Inzulinadagoló/Ellenőrző szoftver/SRS/3.3.2

Funkció	Inzulinadag kiszámítása.
Leírás	Kiszámítja a szállítandó inzulinadagot, ha az aktuálisan mért cukorszint a 3 és 7 egység közti biztonságos zónában van.
Bemenet	Aktuális cukorszint leolvasása (r2) és az előző két leolvasás (r0 és r1).
Forrás	Az aktuális cukorszint leolvasása az érzékelőből. Egyéb leolvasások a memóriából.
Kimenet	CompDose – a szállítandó inzulinadag.
Cél	Fő vezérlőciklus.
Művelet:	A CompDose nulla, ha a vércukorszint stabil, vagy esik, vagy ha a szint növekszik ugyan, de a növekedés üteme csökken. Ha a szint növekszik és a növekedés üteme is nő, akkor a CompDose az aktuális vércukorszint és az előző szint különbségének negyede lesz, kerekítve. Ha az eredmény nullára kerekítődik, akkor a CompDose a szállítható minimális adagra lesz beállítva.
Igény	Két előző leolvasás azért, hogy a cukorszint változásának ütemét ki lehessen számítani.
Előfeltétel	Az inzulinartály legalább a megengedett legnagyobb egyszeri adagnyi inzulint tartalmaz.
Utófeltétel	r0 ki van cserélve r1-re, azután r1 r2-re.
Mellékhatás	Nincs.

6.12. ábra. Rendszerkövetelmények specifikációja szabványos űrlapok segítségével

Az inzulinadagoló a használójának inzulinszükségletét a vércukorszintje változásának üteméből számítja ki, ezt pedig a jelenlegi és a megelőző leolvasásokból számítja. Az inzulinadagoló specifikációjának teljes változata letölthető a könyv honlapjáról.

Amennyiben szabványos űrlapot használunk a funkcionális követelmények megadására, annak a következő információkat tartalmaznia kell:

1. az éppen megadott funkció vagy egyed leírását;
2. bemeneteit és azt, hogy azok honnan származnak;
3. kimeneteit és azt, hogy azok hova kerülnek;
4. utalást arra, milyen egyéb egyedek használtak (az *igény* részben);
5. a végrehajtandó művelet leírását;
6. funkcionális megközelítés használata esetén egy előfeltételt, amely kijelöli, minek kell teljesülnie a függvény hívása előtt, és egy utófeltételt, amely azt adja meg, minek kell teljesülnie a függvény hívása után;
7. a művelet mellékhatásainak leírását (ha vannak).

A formázott specifikáció használata megszünteti a természetes nyelvű specifikáció problémáinak némelyikét. A specifikáció kevésbé változatos, szerkezete kötöttebb, így a követelmények szervezése hatékonyabb. Ugyanakkor nehéz a

követelményeket egyértelműen megfogalmazni, különösen ha összetett számításokra van szükség. Ez látszik a 6.12. ábrán szereplő leírásban: nem tisztázott, mi történik, ha az előfeltétel nem teljesül.

E probléma megoldására táblázatok és grafikus modellek segítségével további információval egészíthetjük ki a természetes nyelvű követelményeket. Ezek leírhatják, hogyan kell a számításokat elvégezni, hogyan váltakoznak a rendszer állapotai, hogyan érintkeznek a felhasználók a rendszerrel, és hogyan hajthatók végre bizonyos műveletsorozatok.

A táblázatok különösen hasznosak, amikor több lehetséges eset mindegyikéhez le kell írunk a végrehajtandó műveleteket. A 6.13. ábrán az inzulinadag kiszámításának átdolgozott leírása olvasható.

Feltétel	Művelet
Cukorszint csökken ($r2 < r1$)	CompDose = 0
Cukorszint változatlan ($r2 = r1$)	CompDose = 0
Cukorszint nő, de a növekedés üteme csökken ($(r2 - r1) < (r1 - r0)$)	CompDose = 0
Cukorszint nő, és a növekedés üteme állandó vagy fokozódik ($(r2 - r1) > (r1 - r0)$)	CompDose = round $((r2 - r1)/4)$ Ha a kerekített eredmény = 0, akkor CompDose = MinimumDose

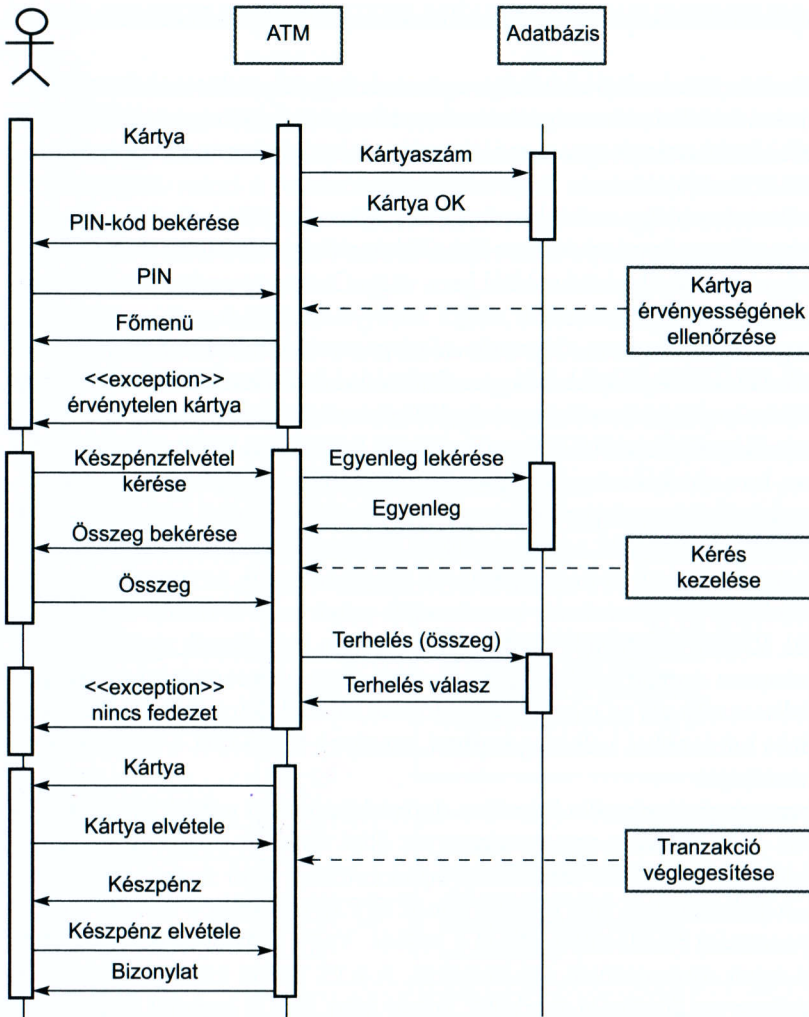
6.13. ábra. A számítás táblázatos megadása

A grafikus modellek akkor a leghasznosabbak, ha az állapotváltozásokat is látnunk kell (8. fejezet), illetve amikor egy műveletsorozatot kell leírunk. A 6.14. ábra azt a műveletsorozatot szemlélteti, ahogy egy felhasználó készpénzt próbál felvenni egy bankautomatából (ATM).

A lezajlott műveletsorozat sorrendjének követéséhez a szekvenciadiagramokat felülről lefelé kell olvasnunk. A 6.14. ábrán három alapvető részsorozat szerepel:

1. *Kártya érvényességének ellenőrzése.* A felhasználó kártyájának érvényessége a kártyaszám és az illető PIN-kódjának ellenőrzésével dől el.
2. *Kérés feldolgozása.* A felhasználó kérését a rendszer feldolgozza. Készpénzfelvétel esetén le kell kérdezni az adatbázisból az illető egyenlegét, és a számlát meg kell terhelni a felvett összeggel. Figyeljük meg, hogy kivétel váltódik ki, ha az ügyfél számláján nincs elég pénz.
3. *Tranzakció véglegesítése.* Az ATM visszaadja az ügyfél kártyáját, és miután ő elvette, kiadja a pénzt és a bizonylatot.

Szekvenciadiagramok újra szerepelnek majd a rendszermodellekről szóló 8. fejezetben és az objektumorientált tervezést tárgyaló 14. fejezetben.



6.14. ábra. Bankautomatából történő készpénzfelvétel szekvenciadiagramja

6.4. INTERFÉSZ-SPECIFIKÁCIÓ

Szinte minden szoftverrendszernek együtt kell működnie olyan, meglévő rendszerekkel, amelyeket valamilyen környezetben már megvalósítottak és telepítettek. Ha az új rendszernek és a meglévőknek együtt kell dolgoznia, akkor a meglévő rendszerek interfészeit pontosan kell specifikálni. Ezeket a specifikációkat a folyamat elején szükséges definiálni, és szerepeltetni kell (esetleg függelékként) a követelménydokumentumban.

Három olyan interfész típus létezik, amit szükséges lehet definiálnunk:

1. *Procedurális interfészek*, ahol a létező programok vagy alrendszerek interfész eljárások hívásával elérhető szolgáltatások sorát nyújtják. Ezeket az interfészeket sokszor alkalmazás programozói interfésznek (Application Programming Interface, API) is hívják.
2. Az egyik alrendszerből a másikhoz átadott *adatszerkezetek*. A (8. fejezetben ismertetett) grafikus adatmodellek az ilyen leírások legjobb jelölései. Amennyiben szükséges, ezekből a leírásokból Java vagy C++ programleírások is automatikusan generálhatók.
3. A már meglévő rendszerre alapozott *adatrepresentációk* (mint bitsorozatok). Ezek az interfészek leginkább beágyazott, valós idejű rendszerekben fordulnak elő. Némely programozási nyelv, például az Ada (a Java nem) támogatja az ilyen szintű specifikációt. Ugyanakkor ezek leírásának legjobb módja valószínűleg az, ha a struktúra egy diagramját használjuk, amelyben feljegyzésekkel magyarázzuk meg az egyes bits csoportok jelentését.

A 10. fejezetben tárgyalt formális jelölések lehetővé teszik az interfészek egyértelmű definícióját, de specializált természetük miatt nem érthetők meg külön tanulás nélkül. Ritkán használják őket a gyakorlatban interfészek specifikációjára, bár véleményem szerint ideálisak erre a célra. Egy programozási nyelv, például a Java, felhasználható az interfészek szintaktikájának leírására. Ugyanakkor ezt még további leírásokkal kell kiegészíteni, amelyek megadják a definiált műveletek szemantikáját.

A 6.15. ábra egy procedurális interfész-definícióra mutat példát Javában. Ebben az esetben az interfész a nyomtatószerver által nyújtott procedurális interfész. Ez kezeli az állományok különböző nyomtatókra való nyomtatási kérélmének sorát. A felhasználók megvizsgálhatnak egy nyomtatóhoz tartozó sort, és törölhetik nyomtatási feladataikat abból a sorból. Vagy akár át is irányíthatnak munkákat az egyik nyomtatóról egy másikra. A 6.15. ábrán szereplő specifikáció a nyomtatószerver absztrakt modellje, amely nem árul el semmit az interfész részleteiről. Az interfészműveletek funkcionalitását definiálhatjuk strukturált természetes nyelvvel vagy táblázatos leírással.

```
interface PrintServer {  
    // absztrakt nyomtatószervert definiál  
    // igény: Printer és PrintDoc interfész  
    // szolgáltatás: initialize, print, displayPrintQueue, cancelPrintJob, switchPrinter  
    void initialize ( Printer p );  
    void print ( Printer p, PrintDoc d );  
    void displayPrintQueue (Printer p);  
    void cancelPrintJob (Printer p, PrintDoc d);  
    void switchPrinter (Printer p1, Printer p2, PrintDoc d);  
} // PrintServer
```

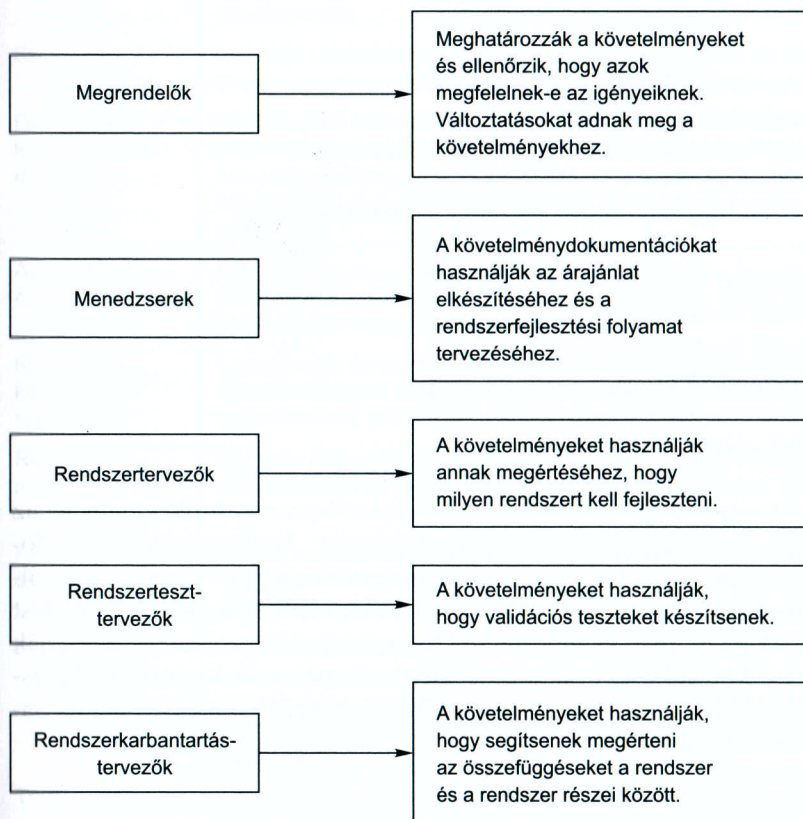
6.15. ábra. Egy nyomtatószerver interfészének leírása Javás PDL-lel

6.5. A SZOFTVERKÖVETELMÉNYEK DOKUMENTUMA

A szoftverkövetelmények dokumentuma (néha hívják szoftverkövetelmény-specifikációnak vagy SRS-nek is) a hivatalos leírása annak, mit kell a rendszerfejlesztőknek megvalósítaniuk. Ajánlott tartalmaznia mind a rendszer felhasználói követelményeit, mind a rendszerkövetelmények részletes specifikációját. Néhány esetben a felhasználói és rendszerkövetelményeket egyetlen leírásba össze lehet fogni. Más esetekben a felhasználói követelményeket a rendszerkövetelmény-specifikáció bevezetéseként adják meg. Ha a követelmények száma magas, a részletes rendszerkövetelményeket külön dokumentumokként készíthetjük el.

A követelménydokumentum használóinak sora igen változatos, a rendszerért fizető szervezet felsőbb vezetőitől a szoftver fejlesztéséért felelős tervezőig. A Gerald Kotonyával írt, követelményelemzésről szóló könyvemből (Kotonya és Sommerville, 1998) átvett 6.16. ábra bemutatja a dokumentum lehetséges használóit és azt, hogy hogyan használják ők a dokumentumot.

A lehetséges felhasználók sokfélesége azt jelenti, hogy a követelmények dokumentumának kompromisszumot kell képeznie a következők között: tájékoztat-



6.16. ábra. A követelménydokumentum használói

nia kell a megrendelőket a követelményekről, a követelmények kellően részletes és pontos definícióját kell adnia a fejlesztők és tesztelők számára, illetve információt kell tartalmaznia a rendszer továbbfejlesztési lehetőségeiről. A várt változásokról szóló információk segíthetik a rendszermérnököket elkerülni a korlátozó tervezési döntéseket, és szintén segíthetnek a rendszert karbantartó mérnököknek hozzáigazítani a rendszert az új követelményekhez.

A fejlesztett rendszer típusától és a használt fejlesztési folyamattól függ, milyen szintű részleteket érdemes tartalmaznia a követelmények dokumentumának. Ha a rendszert egy kívülálló vállalkozó fogja fejleszteni, akkor a kritikus rendszer-specifikációnak pontosnak és nagyon részletesnek kell lennie. Ha a követelményekben nagyobb a rugalmasság, és házon belüli, iteratív fejlesztési folyamat megy végbe, akkor a követelmények dokumentuma kevésbé részletes is lehet, és a többértelműségek a rendszer fejlesztése során is feloldhatók.

Számos nagy szervezet, mint a US Department of Defense és az IEEE, definiált szabványokat a követelménydokumentumok számára. Davis (Davis, 1993) tárgyal néhányat ezek közül, és összehasonlítja a tartalmukat. A legismertebb szabvány az IEEE/ANSI 830-1998-as (IEEE, 1998). Ez az IEEE-szabvány a követelménydokumentumoknál a következő szerkezetet javasolja:

1. Bevezetés

- 1.1. A követelménydokumentum célja
- 1.2. A termék felhasználási területe
- 1.3. Definíciók, betűszavak és egyéb rövidítések
- 1.4. Hivatkozások
- 1.5. A dokumentum hátralevő részének áttekintése

2. Általános leírás

- 2.1. A termék áttekintése
- 2.2. A termék funkciói
- 2.3. Használati jellemzők
- 2.4. Általános megkorlátások
- 2.5. Feltételezések és függőségek

3. **Speciális követelmények**, amelyek a funkcionális, a nemfunkcionális és az interfészekre vonatkozó követelményeket takarják. Nyilvánvalóan ez a dokumentum leglényegesebb része, de a szervezetek gyakorlatának nagy változékonysága miatt nem helyénvaló ennél a résznél is szabványos felépítést definiálni. A követelmények dokumentálhatnak külső interfészeket, leírhatják egy rendszer funkcióit és működését, meghatározhatnak logikai adatbázis-követelményeket, tervezési megkorlátásokat, felmerülő rendszertulajdonságokat és minőségi jellemzőket.

4. Függelékek

5. Tárgymutató

Bár az IEEE-szabvány nem ideális, rengeteg jó tanácsot fogalmaz meg arra vonatkozóan, hogyan írjunk követelményeket és hogyan kerülünk el esetleges problémákat. Ahhoz túl általános, hogy önálló szervezési szabvány legyen. Ez egy általános keretrendszer, amely testre szabható úgy, hogy egy adott szervezet igényeihez igazodó szabványt definiáljon. A 6.17. ábra egy követelménydokumentum lehetséges szervezését mutatja be, amely az IEEE-szabványon alapul. Ugyanakkor ezt kibővítettem úgy, hogy a rendszer megjósolt evolúciójáról tartalmazzon információt. Ezt először Heninger javasolta (Heninger, 1980), és ez – ahogy már tárgyaltam – segít a rendszer karbantartóinak, egyúttal lehetővé teszi a tervezőknek, hogy jövőbeli rendszereszközök támogatását is beépítsék.

Fejezet	Leírás
Előszó	Körvonalazza a dokumentum célzott olvasóit, és leírja a verzió történetét, beleértve egy új verzió létrehozásának magyarázatát és az egyes verziókban végzett változtatások összefoglalását.
Bevezetés	Indokolja, miért van szükség a rendszerre. Röviden leírja annak funkcióit, és elmagyarázza, hogyan fog együttműködni más rendszerekkel. Leírja, hogyan illeszkedik a rendszer a szoftvert megrendelő vállalat átfogó üzleti vagy stratégiai céljai közé.
Szójegyzék	Definiálja a dokumentumban használt szakkifejezéseket. Az olvasó tapasztalatát vagy szakértelmét nem szabad feltételeznünk.
Felhasználói követelmények definíciói	A felhasználói és a nemfunkcionális követelmények számára biztosított szolgáltatásokat ebben a fejezetben javallott leírnunk, melynek során használhatunk természetes nyelvet, diagramokat vagy egyéb, a megrendelő számára érthető jelöléseket. A követendő termék- és folyamatszabványokat is célszerű megadnunk.
A rendszer felépítése	A tervezett rendszerfelépítés magas szintű áttekintése, bemutatva, hogyan oszlanak el a funkciók a különböző rendszermodulok között. Az újrafelhasznált architektúrális komponenseket ki kell emelnünk.
Rendszer-követelmény-specifikáció	A funkcionális és nemfunkcionális követelmények részletesebb leírása. Amennyiben szükséges, a nemfunkcionális követelmények további részletekkel egészíthetők ki, például definiálhatunk interfészeket más rendszerekhez.
Rendszer-modellek	Egy vagy több rendszermodellt ad meg, megmutatva a rendszerkomponensek, a rendszer és annak környezete közötti kapcsolatokat. Ezek lehetnek objektummodellek, adatfolyammodellek és szemantikus adatmodellek.
Rendszer-evolúció	Leírja a rendszer alapját képező feltételezéseket, továbbá vázolja a hardver fejlesztése, a változó felhasználói igények stb. következtében bekövetkező előre látható változásokat.
Függelékek	Részletes, specifikus információt kell nyújtani itt a fejlesztett alkalmazásról. A csatolt függelékekre példa a hardver- vagy az adatbázis-leírások. A hardver-követelmények meghatározzák a rendszer minimális és optimális konfigurációját. Az adatbázis-követelmények megadják a rendszer által használt adatok logikai szerkezetét és a köztük lévő kapcsolatokat.
Tárgymutató	A dokumentum több tárgymutatót is tartalmazhat. A hagyományos, betűrend szerinti tárgymutató mellett szerepelhet még a diagramok, funkciók indexe stb.

6.17. ábra. A követelménydokumentum felépítése

Természetesen a követelménydokumentumban nem szereplő információnak a fejlesztett szoftver típusától és a használt fejlesztési megközelítéstől kell függenie. Ha evolúciós megközelítést alkalmaztak egy szoftverterméknél, a követelménydokumentumból számos fentebb javasolt részletes fejezet hiányozni fog. A hangsúly a felhasználói követelmények és a magas szintű, nemfunkcionális rendszerkövetelmények meghatározásán lesz. Ez esetben a tervezők és a programozók saját ítéltéképességüket használják majd annak eldöntésére, hogyan feleljenek meg a rendszer felhasználói követelményeinek.

Ezzel szemben, ha a szoftver egy olyan nagy rendszertervezési projekt része, amely együttműködő hardver- és szoftverrendszereket tartalmaz, gyakran lényeges a követelményeket igen finom részletezettséggel definiálni. Ez azt jelenti, hogy a követelménydokumentumok valószínűleg nagyon hosszúak lesznek, és ajánlatos tartalmazniuk a 6.17. ábrán szereplő fejezetek legtöbbjét, ha nem az összeset. Hosszú dokumentumoknál különösen fontos, hogy tartalmazzanak egy átfogó tartalom- és címszójegyzéket azért, hogy az olvasók megtalálják azt az információt, amire szükségük van.

A követelmények dokumentumai különösen akkor lényegesek, amikor a szoftverrendszert külső vállalkozó fejleszti. Ugyanakkor az agilis fejlesztési módszerek azzal érvelnek, hogy a követelmények olyan gyorsan változnak, hogy a követelmények dokumentuma már akkor elavult, amikor megírták, így az érte tett erőfeszítés javarészt hiábavaló volt. A formális dokumentumok helyett egyes megközelítések, mint az extrém programozás (Beck, 1999), azt javasolják, hogy a felhasználói követelményeket folyamatosan gyűjtsük, és írjuk kártyákra. A felhasználó ezek után a követelményeket sorrendbe állítja aszerint, hogy melyik legyen megvalósítva a következő rendszerfejlesztési iteráció során.

A labilis követelményekkel rendelkező üzleti rendszereknél, úgy gondolom, ez jó megoldás. Ugyanakkor szerintem továbbra is hasznos megírni támogatásként egy rövid dokumentumot, amely rögzíti a rendszer üzleti és megbízhatósági követelményeit. Könnyű elfelejteni a rendszer egészére vonatkozó követelményeket, ha mindig a következő rendszerkiadás funkcionális követelményeire összpontosítunk.

Kulcsfogalmak

- A szoftverrendszerrel szemben támasztott követelmények kijelölik, mit kell tennie a rendszernek, és megkorításokat határoznak meg annak működésére és megvalósítására vonatkozóan.
- A funkcionális követelmények a rendszer által nyújtandó szolgáltatások ismertetései, vagy annak leírásai, hogyan kell bizonyos számításokat elvégezni. A szakterületi követelmények olyan funkcionális követelmények, amelyek az alkalmazás szakterületének jellegzetességeiből származnak.
- A nemfunkcionális követelmények megkorítást jelentenek a fejlesztett rendszerre és a használandó fejlesztési folyamatra. Ezek lehetnek termék-, szerkezeti vagy külső követelmények. Gyakran összefüggésben vannak a rendszer eredendő tulajdonságaival, ezért a rendszerre mint egészre vonatkoznak.

- A felhasználói követelményeket azoknak az embereknek szánják, akik a rendszert megvásárolják és használják. Az érthetőség miatt érdemes természetes nyelven, könnyen érthető táblázatok és diagramok segítségével megírni őket.
- A rendszerkövetelmények feladata, hogy – precíz módon – közöljék a rendszer által teljesítendő funkciókat. A többértelműség csökkentése végett megírhatók egy természetes nyelv valamilyen strukturált formájában, táblázatokkal és rendszermodellekkel kiegészítve.
- A szoftverkövetelmények dokumentuma a rendszerkövetelmények közös célt szolgáló leírása. Úgy kell felépíteni, hogy a rendszer vevői és a szoftverfejlesztők számára egyaránt használható legyen.
- A követelmények dokumentumairól szóló IEEE-szabvány hasznos kiindulópont a specifikusabb követelményspecifikációs szabványok felé.

További irodalom

Software Requirements, 2nd ed. A könyvet követelmények írói és olvasói számára tervezték, és a helyes követelménytervezési gyakorlatot tárgyalja. (Weigers, K. M., 2003, Microsoft Press.)

Mastering the Requirements Process. Ez a jól megírt, könnyen érthető könyv egyetlen módszeren (VOLERE) alapszik, de rengeteg hasznos általános példát tartalmaz a követelménytervezéshez. (Robertson, S. és Robertson, J., 1999, Addison-Wesley.)

Requirements Engineering: Processes and Techniques. Ez a könyv a követelménytervezési folyamat minden aspektusát lefedi és tárgyalja az egyes követelményspecifikációs technikákat. (Kotonya, G. és Sommerville, I., 1999, John Wiley & Sons.)

Software Requirements Engineering. A követelménytervezésről szóló cikkgyűjtemény, amely számos idevágó cikket magában foglal, beleértve a *Recommended Practice for Software Requirements Specification* címűt, amely az IEEE követelménydokumentumokra vonatkozó szabványát tárgyalja. (Thayer, R. H. és Dorfman, M. (eds.), 1997, IEEE Computer Society Press.)

Feladatok

- 6.1. Nevezzen meg és röviden ismertessen négy követelménytípust számítógépes rendszerekre.
- 6.2. Fejtse ki a természetes nyelv használatának problémáit a felhasználói és rendszerkövetelmények használatakor, és rövid példák segítségével mutassa meg, hogyan segítheti a természetes nyelv űrlapokba szervezése néhány probléma elkerülését ezek közül.
- 6.3. Fedezzen fel kétértelműségeket vagy hiányosságokat a következő követelményleírásban, amely egy jegykiadó rendszer részét képezi:

Egy automatizált jegykiadó rendszer vasúti menetjegyeket árul. A felhasználók kiválasztják az úti célt, behelyeznek egy hitelkártyát és megadnak egy személyi azonosító számot. Az automata kiadja a menetjegyet, és hitelkártyájuk számlaegyenlege megterhelődik. Ha a felhasználó megnyomja a start gombot, aktiválódik a lehetséges célállomások menüje és egy, a felhasználónak szóló üzenet, hogy válasszon úti célt. Ha már választott célt, az automata megkéri a felhasználót, helyezze be a hitelkártyáját. Ellenőrzi az érvényességét, majd megkéri a felhasználót, hogy adja meg személyi azonosítóját. Ha a tranzakciót jóváhagyta, kiadja a jegyet.

- 6.4. Írja át a fenti leírást az ebben a fejezetben leírt strukturált megközelítés segítségével. Oldja fel a felismert kétértelműségeket valamilyen megfelelő módon.
- 6.5. Rajzoljon szekvenciadiagramot, amely bemutatja a jegykiadó rendszerben végrehajtott műveleteket. Bármilyen ésszerű feltételezés megtehető a rendszerről. Szenteljen különös figyelmet a felhasználói hibáknak.
- 6.6. A javasolt technikát használva, ahol a természetes nyelv a szokásos módon jelenik meg, írjon a valóságba illő felhasználói követelményeket a következő funkciók számára:
 - Egy banki ATM pénzkidó funkciója.
 - A helyesírás-ellenőrző és -javító funkció egy szövegszerkesztőben.
 - Egy felügyelet nélküli benzinkúti rendszer, amely hitelkártya-leolvasót tartalmaz. A vásárló végighúzza a kártyát a leolvasón, majd megadja az üzemanyag kívánt mennyiségét. Az üzemanyag kitöltődik, és az összeg a vevő számlájára terhelődik.
- 6.7. Írja le a rendszerre vonatkozó nemfunkcionális követelmények négy különböző típusát. Adjon példákat mindegyik fajta követelményre.
- 6.8. Írjon egy listát nemfunkcionális követelményekből a jegykiadó rendszerhez, kiemelve az elvárt megbízhatóságot és a rendszer válaszüdejét.
- 6.9. Tegyen javaslatot arra, hogyan követheti nyomon a funkcionális és nemfunkcionális követelmények közötti kapcsolatokat az a fejlesztő, aki a rendszerkövetelmény-specifikáció kidolgozásáért felelős.
- 6.10. Ön elvállalt egy állást egy szoftverfelhasználónál, aki az ön korábbi munkaadójával kötött szerződést egy rendszer kifejlesztésére. Felfedezi, hogy az ön jelenlegi vállalata másképpen értelmezi a követelményeket, mint a korábbi munkaadója. Fejtse ki, mit javallott tennie ilyen helyzetben. Azt is tudja, hogy a jelenlegi munkaadójára háruló költségek növekedni fognak, ha a kétértelműséget nem oldja föl. Az előző munkaadójának viszont titoktartási felelősséggel tartozik.