Megbízhatóság

A szoftvernek, a szoftverrendszerhez tartozó programoknak tartósan hibátlanul kell működniük. Nem nehéz megítélni a követelmény erejét, ha a mai személyi számítógépeken futó programok leállásának gyakoriságára gondolunk. A követelmény teljesülésének mércéje lehet például a szoftver egymást követő két leállása között eltelt időtartam.

Helyesség

A szoftver a követelményekben meghatározott feladatokat tökéletesen ellátja, a kijelölt feldolgozásokat, műveleteket hibátlanul elvégzi, pontos adatokat szolgáltat. Hozzá kell tenni, hogy a program helyes működésének bizonyítása nem egyszerű feladat. Még a legszigorúbb, módszeresen megtervezett tesztelés mellett is előfordulhat, hogy az üzembe helyezés után évek múlva derül fény egy addig soha nem észlelt hibára.

Felhasználóbarátság

A szoftver legyen ergonómiai szempontból átgondolt, könnyen kezelhető. Objektív mércével nehezen leírható, de alapvetően fontos, egyre inkább központi kérdésként kezelendő kritérium. Alapvető elvárás, hogy a szoftver legyen informatikai jártasság nélkül megtanulható, kezelhető. A felhasználói felület grafikái segítsék a tájékozódást, a dokumentációk maximálisan támogassák a szoftverrel végzett munkát.

Hibatűrés

A szoftvert fel kell készíteni a környezetében felmerülő lehetséges hibákra. A szoftver nem elszigetelten, önmagában működő rendszer, folyamatosan kapcsolatban áll a külvilággal, más környező rendszerekkel. A fejlesztőknek át kell gondolniuk azokat a váratlan, külső kapcsolatok következményeként bekövetkező eseményeket, amelyek befolyásolhatják a szoftver működését. (Például: nyomtatás közben nem áll le, hanem figyelmeztet a hiba lehetséges okára.)

Hatékonyság

A szoftvert úgy kell elkészíteni, hogy az maximálisan használja ki az adott környezet szoftver és hardver erőforrásait. A tervezés során át kell gondolni a szoftver teljesítményét, működési sebességét befolyásoló maximális terhelések paramétereit, és a szélsőséges helyzetek (például a feldolgozandó adatmennyiség megnövekedése) kezelésére alkalmas takarékos megoldásokra kell törekedni.

Bővíthetőség, rugalmasság

A szoftvernek alkalmasnak kell lennie arra, hogy az informatikai háttér fejlődését követni tudja. (Például: ne kelljen a szoftvert teljesen átírni, ha egy új operációs rendszer verzió jelenik meg.)

Újrafelhasználhatóság

A gazdaságossági szempontok megkövetelik, hogy a szoftvert vagy annak részeit többször, más-más fejlesztésekben újra felhasználhassuk, a moduláris felépítésű szoftver moduljait új rendszerek fejlesztése során beépíthessük.

Hordozhatóság

A fejlesztőknek arra kell törekedniük, hogy a szoftver legyen futtatható többféle szoftver és hardverkörnyezetben. Ez a kritérium természetesen nem teljesíthető végletekig, de el kell érni, hogy a szoftver a lehető legnagyobb mértékben független legyen az hardvereszközöktől és az operációs rendszertől.

Együttműködési képesség

A szoftvernek nyitottnak kell lennie más rendszerek felé, alapvető elvárás, hogy igény esetén együtt tudjon működni más alkalmazásokkal. Biztosítson lehetőséget például arra, hogy a feldolgozás eredményeként kapott kimeneteket más szoftver által feldolgozható formában exportálja. Ez a kritérium abban az esetben is nagyon fontos, ha a meglévő rendszert új szolgáltatásokkal bővíteni szeretnénk.

Jó és hibás program összehasonlítása

A programokat emberek írják, ezért természetesnek mondható, hogy a kódban sok hiba fordul elő. Ezeket a hibákat többnyire figyelmetlenség okozza, de gyakran előfordul, hogy a programozó hozzá nem értéséből adódnak. Az alábbiakban pár példával szeretném szemléltetni a jó és a hibás programok közötti különbséget a fentiek alapján.