Johanyák, Zs. Cs.: Hibalehetőség és hibahatás elemzés alkalmazása a szoftverfejlesztésben, Informatika a Felsőoktatásban 2002, Debrecen, 2002. augusztus 28-30. pp. 833-840. ISBN 963 472 691 7 http://johanyak.hu

HIBALEHETŐSÉG ÉS HIBAHATÁS ELEMZÉS ALKALMAZÁSA A SZOFTVERFEJLESZTÉSBEN

APPLYING FAILURE MODE AND EFFECTS ANALYSIS IN SOFTWARE ENGINEERING

Johanyák Zsolt Csaba, johanyak.csaba@kefo.hu
Kecskeméti Főiskola Műszaki Főiskolai Kar

Abstract

The aim of the FMEA is the recognition of different failure possibilities, and related risks in a very early phase of the products life-cycle, the prevention of failure occurrence, and the elimination of possible failures, to achieve direct cost savings and to maintain the good reputation of the company. This method is applicable not only during the design phase but for working systems too.

During the analysis a team of specialists searches for the most frequently occurring failures, the ones which cause most severe consequences, and the most rarely verified points. The team makes proposals for the failure prevention and risk reduction, and possibly improvement of the efficiency of verification. The execution of the proposals is controlled and evaluated regularly. Through the application of FMEA there are possibilities to develop a controlling system which guarantees the failure recognition and prevention, and the continuous quality improvement. This paper presents the application of the FMEA in the field of software engineering, and gives a survey of the most important concepts of this topic.

Összefoglaló

A hibalehetőség és hibahatás elemzés célja az egyes hibalehetőségek és a hozzájuk kapcsolódó kockázatok felismerése a termék életciklusának minél korábbi szakaszában, a hiba előfordulásának megelőzése és az esetlegesen fellépő hibák felhasználóhoz való eljutásának megakadályozása, valamint közvetlen költségmegtakarítás elérése és a vállalat jó hírnevének megőrzése. A módszer nemcsak a fejlesztés során, hanem már működő rendszerek esetén is alkalmazható.

Az eljárás során a szakértőkből alakított munkacsoport megkeresi a leggyakrabban előforduló, legsúlyosabb következményekkel járó és a leggyengébben ellenőrzött hibákat, majd javaslatot készít azok megelőzésére, súlyosságának csökkentésére, esetleg az ellenőrzés hatékonyságának javítására. A csoport rendszeresen ellenőrzi javaslatainak végrehajtását és hatását. A módszer alkalmazása révén lehetőség nyílik egy olyan szabályozó rendszer kialakítására, mely garantálja a hibák felismerését, rendszeres kiküszöbölését, és ezzel az egyre jobb minőségű termékek előállítását. Előadásom célja a módszer szoftverfejlesztés területén való alkalmazásának vizsgálata és a témakör sajátosságainak megfelelő fogalomértelmezések áttekintése.

HIBALEHETŐSÉG ÉS HIBAHATÁS ELEMZÉS ALKALMAZÁSA A SZOFTVERFEJLESZTÉSBEN

Johanyák Zsolt Csaba, johanyak.csaba@kefo.hu
Kecskeméti Főiskola Műszaki Főiskolai Kar

Az informatika széleskörű térhódításával párhuzamosan egyre erősebb igény mutatkozik a piac részéről a szoftvertermékek minősége iránt. A piaci elvárások növekedése, a szoftverrendszerek egyre komplexebbé válása és az erős konkurencia hatására csak azok a fejlesztő cégek tudnak talpon maradni hosszabb távon, akik jól működő – nemcsak papíron létező – minőségügyi rendszert építenek ki, és munkájuk során alkalmazzák a gazdaság különböző területein már jól bevált minőségjavító technikákat a szoftverfejlesztés területén.

Előadásom célja a hibalehetőség és –hatás elemzés szoftverfejlesztés területén való alkalmazásának ismertetése, a témakör sajátosságainak megfelelő fogalomértelmezések áttekintése és a kockázati mérőszámok megállapítási módjainak meghatározása.

1. Történeti háttér

Az FMEA-t (Failure Mode and Effects Analysis) az ötvenes években fejlesztették ki az űrhajózás területén. A repülőgépipar és az űrhajózás-technika kiváló alkalmazási területet jelentett, hiszen itt olyan berendezések készülnek és működnek, amelyeknél már a legkisebb hiba is emberi áldozatokat követelhet, vagy jelentős anyagi kárt okozhat, ezért magas megbízhatósági szintre van szükség. Ennek következtében kiemelt hangsúlyt fektetnek a gyártás megkezdése előtt a hibalehetőségek kiküszöbölésére.

A módszer igazán széleskörű elterjedése az autóipari alkalmazásnak köszönhető, mivel az autógyártó cégek beszállítóiktól is megkövetelik alkalmazását a General Motors, a Crysler, és a Ford által kidolgozott QS 9000 előírásainak megfelelően.

2. Az elemzés célja

A hibalehetőség és -hatás elemzés célja az egyes hibalehetőségek és a hozzájuk kapcsolódó kockázatok felismerése a termék életciklusának minél korábbi szakaszában, a hiba előfordulásának megelőzése és az esetlegesen fellépő hibák vevőhöz való eljutásának megakadályozása, ezáltal egyrészt közvetlen költségmegtakarítás elérése, másrészt a vállalat jó hírnevének megőrzése.

A szoftver FMEA célja a szoftver architektúra vagy a fejlesztési folyamat átvizsgálása olyan kockázatokra koncentrálva, mint a biztonság és a rendelkezésre állás. A szoftverre végrehajtott elemzés céljainak konkrét megfogalmazása eltérő lehet attól függően, hogy a fejlesztés mely szakaszában kerül sor az FMEA-ra. A módszer nemcsak a fejlesztés során,

1. ábra Az elemzés céljainak változása a fejlesztés egyes szakaszaiban

hanem már működő rendszerek esetén is alkalmazható. Újbóli végrehajtására illetve átvizsgálására minden olyan esetben sor kerül, amikor valamit változtatnak a szoftverben.

3. Az elemzés típusai

Kezdetben az elemzés tárgyának függvényében a módszer két fő típusát a konstrukciós (design) és a folyamat (process) FMEA-t különböztették meg. A módszer népszerűvé válása újabb és újabb altípusok megjelenését eredményezte. Ezek alapvetően a vizsgált terület részekre bontásának módjában valamint a kockázati mérőszám kialakítása során alkalmazott komponens értékek értelmezésében és osztályozási módszereiben térnek el egymástól.

A szoftver előállítása során az alkalmazott fejlesztési modell minden egyes szakaszában alkalmazhatjuk az eljárást, és eszerint osztályozva is különböző altípusokról beszélhetünk, de ezek valójában mind a magyar terminológiában konstrukciósnak nevezett Design FMEA megvalósításai.

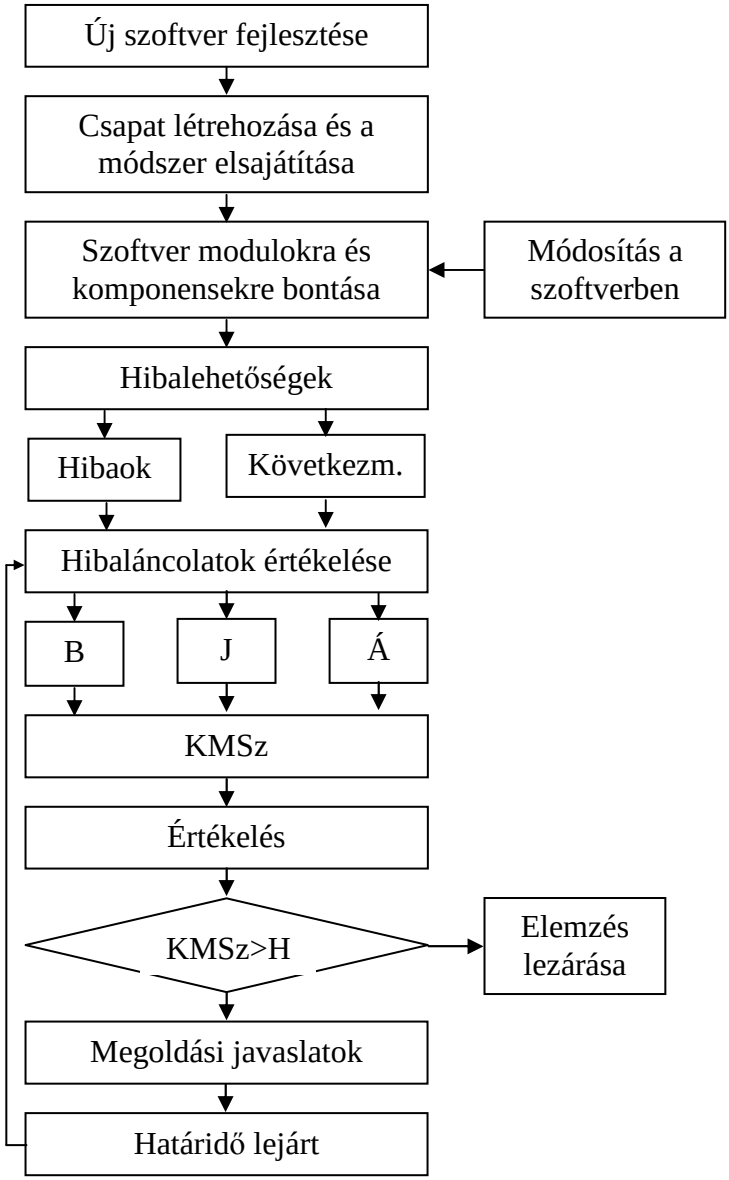
Az elemzési technika szerint a módszer három típusáról különböztetjük meg:

- modul alapú megközelítés: a szoftvert modulokra bontja, és megkeresi ezek hibalehetőségeit;
- feladatközpontú megközelítés: a szoftver funkciói szerint haladva történik a vizsgálat;
- helyzetalapú megközelítés: a szoftver egy lehetséges hibás működéséből kiindulva keresik meg a hibát előidéző rendszerelemeket.

A továbbiakban a modul alapú megközelítéssel foglalkozunk.

4. Az elemzés lépései

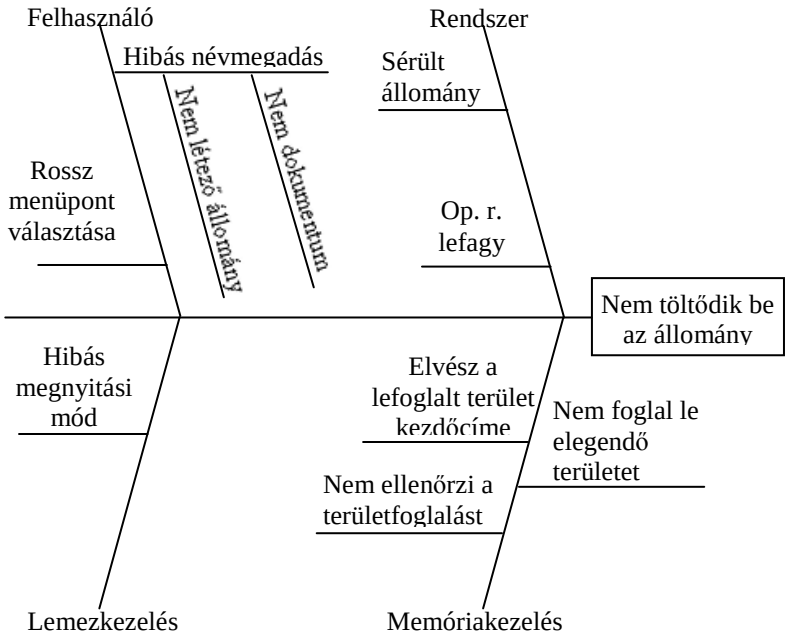
Az eljárás messzemenően formalizált, folyamata hasonló lépések sorozatából épül fel minden típusa esetén (2. ábra). Az elemzést csoportmunkában végzik, így a csapat kialakítása



2. ábra Az elemzés lépései

az első feladat. A vezető az ún. moderátor személye és képességei nagyban meghatározzák a munka hatékonyságát. Feladata az FMEA folyamat ismertetése, majd a későbbiekben az elemzés irányítása. Alapos módszertani ismeretekkel és gyakorlattal kell rendelkezzen. A munkacsoport ideális esetben 4-6 szakemberből áll össze, akik a szoftvertervezés, -fejlesztés, -tesztelés és a vevőszolgálat területén dolgoznak.

Az eljárás első szakaszában a szoftvert modulokra, komponensekre bontják, és ezek kapcsolatrendszerét grafikus ábrázolás segítségével teszik áttekinthetővé. Az ábrázolástechnika az informatika területén jól ismert folyamatábra, struktogram, UML, stb. lehet, a fejlesztés során alkalmazott hagyományos vagy OOP technikáknak megfelelően. Amennyiben a tervben eleve szerepel ilyen dokumentáció, úgy ebben a szakaszban csak a dokumentáció megismerése és megértése a feladat.



3. ábra Halszálka diagram

A vizsgálat során áttekintik minden modul feladatát. A komponensek mindig a föléjük rendelt, őket meghívó modul feladatainak egy részét látják el. Ezután a feladatok ellátásához kapcsolódó hibalehetőségek feltárása következik. A módszer sikerességének kulcsa, hogy a csapattagok korábbi hasonló komponensek fejlesztésénél, alkalmazásánál szerzett tapasztalataik alapján felismerjék a hibalehetőségeket. Ebben a szakaszban a munka hatékonysága és rendszerezettsége olyan módszerek segítségével fokozható, mint a halszálka (Ishikawa) diagram, strukturált ötletroham, formális elemzés, Delphi módszer, stb. A 3. ábrán egy halszálka diagram látható, ami Windows programozás gyakorlaton elkészített egyszerű szövegszerkesztő alkalmazás vizsgálatának részeként keletkezett. Kialakítása során a „Nem töltődik be az állomány” lehetséges hiba okait kerestük. Az okok négy fő csoportba lettek besorolva.

1. táblázat B értékszámok és magyarázatuk

Érték	Magyarázat
1	Valószínűtlen, hogy a hiba bekövetkezik.
2 - 3	A komponens hasonlít egy olyan korábbi komponenstípushoz, amellyel kapcsolatban aránylag ritkán fordult elő ez a hiba.
4 - 6	A komponens hasonlít egy olyan korábbi komponenstípushoz, melynél alkalmilag előfordult ez a hiba.
7 - 8	A komponens hasonlít egy olyan korábbi komponenstípushoz, melynél sok nehézséget okozott ez a hiba.
9 - 10	Szinte biztos, hogy bekövetkezik a hiba.

Minden egyes hibalehetőség esetén a csapat törekszik az előfordulásban közre játszó összes ok valamint a hiba lehetséges következményeinek felderítésére. A modulok és komponenseik hierarchikus kapcsolatrendszeréből következően gyakran egy komponens hibájának következménye a hierarchiában felette levő (őt meghívó) modul hibás működése lesz, más szóval a modul hibájának okaként az említett komponens hibáját nevezhetjük meg. A hibaláncolatoknak nevezett ok-hiba-következmény hármasok kapcsolatrendszerének feltárása nagy figyelmet igényel, amiben jelentős segítséget nyújthat egy egyszerűsített hibafa felállítása. A munka dokumentálása során a hibaláncolatok azonosítására egy pontokkal elválasztott csoportokból álló egyszerűsített decimális jelölésrendszer használható. Használatát egy példán keresztül vizsgáljuk meg. Az 1.2.1.3.1 jelentése: 1. modul, 2. komponens, a komponens 1. hibalehetősége, a hiba 3. következménye, a hiba 1. oka.

A hibaláncolatok felismerését és dokumentálását követően minden modul-komponens-hiba-következmény négyest három szempont szerint minősít a csoport 1 és 10 közötti értékszámokkal. A minősítés alapja a korábbi hasonló esetekben szerzett tapasztalat vagy ennek hiányában a becslés, így az eredményben szükségszerűen megjelennek szubjektív elemek, bár a problémamegoldás csoportos jellege miatt ezek hatása nem jelentős. Léteznek autóipari illetve általános irányelvek a minősítés meghatározására, de ha a csoport gyakran végez FMEA-t, akkor egy kis idő után kialakul egy saját értékrendszer, amit a minőségügyi rendszer elvárásainak megfelelően formában rögzíthetnek. A három értékelési szempont:

- a bekövetkezés valószínűsége (B)
- a hiba jelentősége (J)
- annak a valószínűsége, hogy a hibát nem ismeri fel a tervezett tesztelés (Á)

A bekövetkezés valószínűségét minősítő értékszámok meghatározásához nyújt segítséget a [4] adaptációjával készült 1. táblázat. A mérőszámok megállapítása során figyelembe veszik a tesztelési tervet és minden olyan körülményt, ami megelőzheti a hiba bekövetkezését vagy csökkentheti annak jelentőségét. Ezután a B, J és Á értékek szorzataként a csoport meghatározza a kockázati mérőszámot (KMSz), amit egy Pareto elemzés keretében a modul-komponens-hiba-következmény négyesek fontossági sorrendjének megállapítására használnak fel. A rangsorolást követően a KMSz értekek szerint csökkenő sorrendben haladva a csoport javaslatokat tesz a felismert hibák kiküszöbölésére, bekövetkezésük megelőzésére vagy a tesztelés kiterjesztésére. Minden megoldási javaslathoz megvalósítási határidőt és felelős személyt rendelnek, akinek feladata a részletes kidolgozás és gyakorlatba ültetés.

Az előre megszabott határidők leteltével a csoport felülvizsgálja az érintett hibaláncolatokat megvizsgálva a javasolt megoldás alkalmazását és hatékonyságát. A visszatérő elemzés során újból meghatározzák a B, J, Á és KMSz értékeket, és ennek alapján döntés születik arról, hogy a szoftvertervben végrehajtott módosítások elfogadható mértékűre csökkentették-e a kockázati mérőszámot vagy további intézkedésekre van szükség.

A csoport értékelési rendszerének kikristályosodása után meghatározhatnak egy általános kockázati mérőszám határértéket (H), ami alatt eleve nem foglalkoznak a felismert hibalehetőségekkel, illetve a visszatérő elemzés során az ez alatti KMSz értékeknél a problémát megoldottnak tekintik. Ezt a határértéket általában csak akkor veszik figyelembe már az első elemzés során, ha a vizsgálatra kerülő hibaláncolatok száma igen magas. Az FMEA alkalmazását megkövetelő megrendelők legtöbbször azt várják el, hogy a módszert

alkalmazó cég minőségjavító tevékenységének eredménye abban is nyilvánuljon meg, hogy folyamatosan csökkentik a határértéket.

Az elemzés akkor tekinthető lezártnak, ha minden kockázati mérőszámot sikerült a határérték alá csökkenteni. A továbbiakban minden olyan esetben, ha a szoftverben valamilyen módosítást hajtanak végre, akkor felülvizsgálják, és szükség szerint kiegészítik, újratárgyalják a szoftverhez készült FMEA-t.

5. Dokumentálás

Az eljárás nyomon követhetőségét egy jól áttekinthető táblázatos dokumentálási forma segíti. Ez elkészíthető akár egy szövegszerkesztő vagy táblázatkezelő programban, de léteznek az elemzést célirányosan támogató szoftverek is. A táblázat fejléce két részből épül fel. A felső öt sor a módszert alkalmazó cég szervezeti és minőségügyi rendszer sajátosságait tükrözi, tartalmazza a vizsgált szoftver azonosításához szükséges információkat, oldalszámot, az elemzésben jelentős szereppel bíró személyek megnevezését és aláírását. A fejléc alsó része az elemzés végrehajtása során keletkező információk számára szükséges oszlopokat határozza meg.

2. táblázat FMEA űrlap

Szoftver FMEA					Vizsgált szoftver: szövegszerkesztő					Lapszám: 1						
					Felelős személy:					Azn. jel: 1./2002.						
Felelős ter:					Készítette:	Dátum:										
Érintett ter.					Átdolgozta:	Dátum:										
Tervező:					Jóváhagyta:	Dátum:										
					Jelenlegi állapot							Javított állapot				
Modul	Komponens	Hiba	Következmény	Ok	Ellenőrző intézkedések	B	J	Á	KM Sz	Javaslat	Felelős /Határidő	Végrehajtott	B	J	Á	KM Sz
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1	1. Menürendszer	1. Hiányos	1. Funkciók hiánya	1. Fejlesztés időhiánya	Az idő előkalkulációja	3	6	1	18	Újratervezés						
		2. Nem működik megfelelően	1. Adatvesztés	1. Hibás forráskód	Tesztelés	4	8	1	32	Forráskód javítás						

Minden hibaláncolat esetén az első két oszlop alapján azonosítható be a vizsgálat aktuális tárgya, a 3. - 5. oszlopok határozzák meg a hibaláncolatot, a 6. - 10. oszlopok a jelenlegi állapotot tükrözik, a 11. – 12. oszlopok a probléma megoldási módjára utaló információkat tartalmaznak, míg a 13. - 17. oszlopok kitöltésére a visszatérő elemzés alatt kerül sor.

6. Irodalomjegyzék

[1] DIN 25448:1990 Ausfalleffektanalyse (Fehler-Möglichkeits- und -Einfluß-Analyse)

[2] MIL-STD-1629A: 1980 Procedures for performing a Failure Mode, Effects and Criticality Analysis.

[3] Schubert, M.: FMEA – Fehlermöglichkeits- und Einflußanalyse. Leitfaden, Deutsche Gesellschaft für Qualität e.V., Frankfurt am Main, 1993.

[4] Ford Q 101