# ELIMINÁCIA RIZÍK POMOCOU MOŽNÝCH

*Vo všetkom, čo robíš, je riziko, ale bez rizika by si nič nedosiahol. Walter Scheel*

*Peter Paššák*

Slovenská technická univerzita
Fakulta informatiky a informačných technológií
Ilkovičova 3, 842 16 Bratislava
peterpassak[zavináč]gmail[.]com

***Abstrakt.*** *Kvalita softvéru a hlavne jeho bezpečnosť sa stávajú čoraz dôležitejšími cieľmi pri jeho vývoji. To samozrejme platí aj pre požiadavky zákazníkov, ktorí chcú spoľahlivé a bezpečné systémy. Metóda stromu poruchových stavov je jedna z metód používaná na odhaľovanie rizikových miest v zložitých systémoch a na zvyšovanie ich bezpečnosti. Esej za zaoberá použitím tejto metódy v rôznych fázach životného cyklu softvéru. Je v nej popísaný vznik a vývin tejto metódy, postup, ktorým sa daný strom poruchových stavov tvorí a časti, z ktorých môže byť zložený. Táto metóda sa dá použiť aj pri komunikácií so zákazníkom, kde má isté prínosy, čo je tiež spomenuté*

***Kľúčové slová:*** *strom poruchových stavov, FTA*

## Úvod

Riadenie projektov. Je to činnosť, ktorú majú v plnej miere na starosti ľudia, manažéri. O aké projekty v tomto prípade ide, nás až tak nemusí zaujímať, pretože pri žiadnom sa nevyhneme jednej veci a to sú chyby alebo chybné rozhodnutia. Ako hovorí jedna stará múdrosť, mýliť sa je ľudské a pri riadení projektov môže táto ľudská vlastnosť pôsobiť niekedy katastrofálne. Pre odvrátenie takýchto prípadov vznikol druh manažmentu známy ako manažment rizík. Úlohou tohto manažmentu je identifikovať možné riziká, analyzovať ich a definovať optimálny spôsob ich zvládnutia pri minimálnych nákladoch a rešpektovaní stanovených cieľov. Ďalšou úlohou manažmentu rizík je predovšetkým dosiahnutie maximálnej možnej bezpečnosti a ochrany majetku vypracovaním optimálnej stratégie riadenia rizík, ktoré sú potenciálnym zdrojom budúcich škôd.

Pri riadení projektov teda hrajú dôležitú úlohu ľudské skúsenosti a znalosti. Aj vďaka nim sa vedia skúsení manažéri vyhnúť veľkému množstvu chýb. Toto platí samozrejme pre všetky typy projektov, no my sa budeme zaoberať projektmi softvérovými. Kontrola rozhodnutí môže mať tým pádom veľký význam pri zlepšení kvality a spoľahlivosti softvéru, keďže pre zložitosť ľudského správania, nie je možné odstránenie všetkých ľudských chýb v rozhodovaní. Pre toto je potrebné analyzovať rozhodnutia, aby sa včas odhalili chyby, ktoré by nakoniec mohli mať fatálne následky na celkový výsledok.

Väčšina príčin, na ktorých môže zlyhať projekt, je spôsobená zlým rozhodnutím človeka. V tomto prípade sa nejedná iba o chyby, ku ktorým môže dôjsť počas vývoja produktu, ale aj o aktivity pred začiatkom vývoja. Jedná sa napríklad o zabezpečenie hardvéru proti krádeži, či výber kvalitných a spoľahlivých zamestnancov.

Na analýzu tohto typu existuje viacero metód a jednou z nich je analýza stromom poruchových stavov, ktorú je možné použiť vo viacerých fázach životného cyklu softvéru, ale tiež aj pri komunikácii so zákazníkom namiesto klasických UML diagramov.

## Analýza stromom poruchových stavov

Analýza stromom poruchových stavov (FTA z ang. Fault Tree Analysis) je metóda analýzy spoľahlivosti zložitých systémov, ktoré sa skladajú z funkčne viazaných alebo závislých podsystémov, ktoré slúžia na rôzne účely. Jej princípom je dekompozícia vrcholovej udalosti, čo je poruchový stav, ktorý skúmame a jeho analýzou sa snažíme nájsť podmienky a ich kombinácie, ktoré musia nastať, aby došlo k takejto situácií. FTA sa môže použiť v prípade vývoju softvéru v rôznych situáciách a to buď preventívne najčastejšie pri návrhu, alebo na vyšetrovanie už vzniknutých problémov. Toto je veľmi užitočná vec, pretože sa môže použiť viackrát tá istá metóda počas životného cyklu softvéru, čo môže ušetriť čas potrebný na naučenie sa inej metódy. Problém je v tomto prípade možno v tom, že v každej fáze ju zrejme budú používať iní členovia tímu, ale to nemusí byť pravidlo.

Táto metóda bola predstavená v roku 1962 pre uľahčenie analýzy nespoľahlivosti protiraketovej obrany v USA a konkrétne to bolo pri vývoji bezpečnosti štartovacieho systému rakety Minuteman. Neskôr bola zdokonalená v spoločnosti Boeing a okrem letectva našla rýchlo uplatnenie aj v kozmonautike, zbrojnom priemysle či jadrovej energetike, čiže všade tam, kde bolo potrebné analyzovať zložité systémy a bolo treba zabezpečiť vysoký stupeň spoľahlivosti. Už z tohto je zrejmé, že ide o metódu, ktorá pri správnom použití dokáže odhaliť závažné nedostatky, ktoré by v určitých nepredvídaných situáciách mohli nastať, inak by určite nebola používaná pri takých projektoch.

FTA našla uplatnenie aj pri vývoji softvéru a je možné ju aplikovať v rôznych štádiách jeho životného cyklu. Prvé pokusy aplikovať ju v tejto oblasti sa zamerali na jej použitie na úrovni kódu. Toto použitie je však príliš zložité a zdĺhavé, pretože analýza celého systému iba pomocou analýzy zdrojového kódu, by pri väčších systémoch bola skoro nemožná. Takisto sa v tejto fáze odhalené chyby oveľa ťažšie odstraňujú, hlavne ak sa jedná o chyby v návrhu. Okrem aplikácie metódy FTA na zdrojový kód je možné aplikovať túto metódu aj vo fáze získavania požiadaviek a návrhu, kde v prípade odhalenia chýb dôjde k väčšej úspore času a zdrojov.

Pri použití metódy FTA je potrebné poznať ako systém funguje, alebo bude fungovať. Základné predpoklady majú obsahovať očakávané prevádzkové podmienky a treba aj

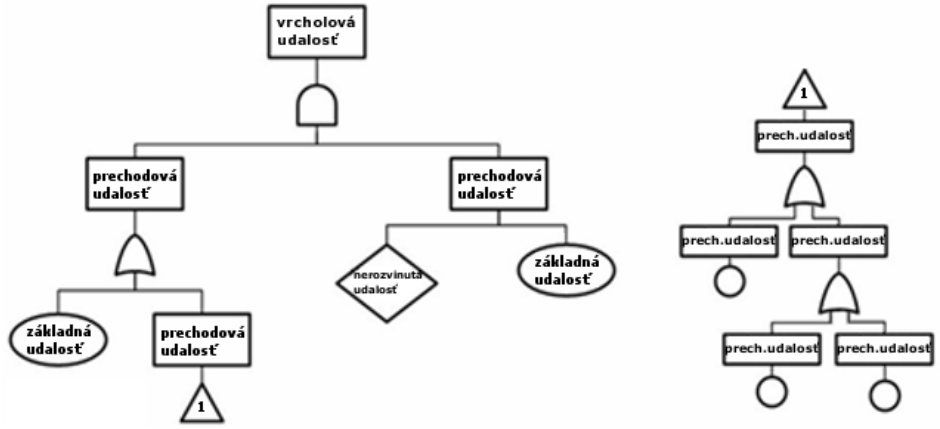
zvážiť výsledky, ktoré je možné z tejto metódy získať, údaje potrebné na vykonanie analýzy, jej komplexnosť, náklady, ktoré sú potrebné na vytvorenie analýzy a tiež ďalšie faktory. Je dôležité vždy zvážiť, o aký projekt ide a či je použitie tejto metódy vhodné. Vytvorenie FTA môže zabrať dlhší čas, ktorý sa však líši aj podľa veľkosti a zložitosti skúmaného systému. Pri používaní tejto metódy sa však dá vychádzať aj z už vytvorených analýz pre iné projekty, čo môže ušetriť veľa času. Myslím, že aj v prípade, keď nejde o zložitý projekt sa môže táto metóda použiť aspoň vo fáze získavania požiadaviek, aby sa predišlo podceneniu projektu [4].

## Tvorba stromu poruchových stavov a jej postup

Pre lepšiu predstavu o tom ako táto metóda funguje si popíšeme postup tvorby stromu poruchových stavov a tiež jednotlivé stavebné prvky. Pri tejto metóde je veľmi dôležité určiť vrcholnú udalosť, čo je vlastne udalosť, ktorej sa chceme vyhnúť a následne pre ňu veľmi zodpovedne zostaviť strom porúch.

Metódu možno rozdeliť na viac štádií, pričom prvým z nich je voľba vrcholnej udalosti a všeobecné identifikovanie možných príčin, kde je možné využiť napríklad skúsenosti z predchádzajúcich projektov. Krokom dva je určenie reťazových porúch, ktoré vedú k vrcholovej udalosti. Tu sa hľadajú spôsoby akými dochádza k poruchám jednotlivých komponentov a záverom by mali byť príčiny porúch, pre každý z nich.

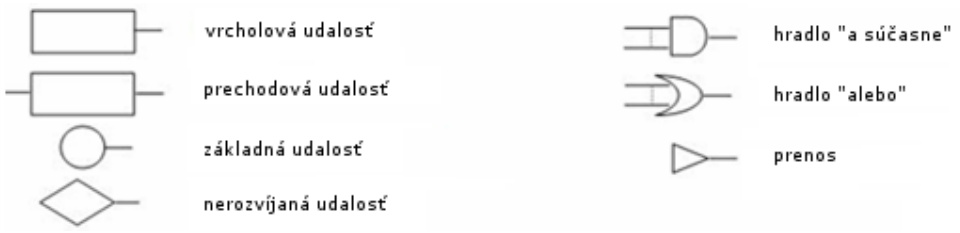
Touto analýzou sa vytvorí dobre čitateľný a systematicky usporiadaný strom, v ktorom je vidieť ako jednotlivé základné prvky prispievajú k poruchám systému. Schému takéhoto stromu môžeme vidieť na obrázku 1 [3,4].




**Obr. 1.** Schéma FTA.

Ako vidieť, základný nástroj tejto analýzy predstavuje grafické vyjadrenie vzťahu medzi jednotlivými čiastkovými udalosťami a konečnou vrcholovou nežiaducou udalosťou. Na jeho zostavenie sa používa ustálená grafická symbolika, ktorú môžeme vidieť na obrázku 2. Takáto forma zápisu je prehľadná a ľahko pochopiteľná. Myslím, že je to hlavná zbraň tejto analýzy, pretože už pri letmom pohľade a prečítaní si názvov jednotlivých symbolov,

začne mať človek určitú predstavu, ako by to mohlo fungovať. Osvojenie si tento metódy určite nezaberie veľa času.




**Obr. 2.** Grafické symboly v FTA.

Grafická symbolika obsahuje niekoľko symbolov, pričom by sme ich mohli rozdeliť na dve základné skupiny. Prvou sú prvky znázorňujúce udalosti rôzneho druhu a druhou hradlá znázorňujúce vzťahy, čiže logické väzby medzi jednotlivými udalosťami. Okrem týchto skupín je tu ešte symbol prenos, čo je akýsi odkaz na pokračovanie vetvy na inom mieste. Jeho úlohou je sprehľadňovať štruktúru stromu a tiež zabrániť duplicite.

Skupina udalostí obsahuje prvky ako vrcholová udalosť, ktorá označuje udalosť, pre ktorú danú analýzu zostrojujeme. Na opačnej strane sa nachádzajú základné udalosti a nerozvíjané udalosti, ktoré predstavujú udalosti, o ktorých nemáme dostatok informácií, alebo ich nepotrebujeme ďalej analyzovať. Medzi nimi zvyknú byť prechodové udalosti.

Skupinu hradiel tvoria dve hradlá a to hradlo „a súčasne“ a hradlo „alebo“. Tieto hradlá predstavujú logický vzťah, ktorý je medzi udalosťami, ktoré do hradla vchádzajú a podľa toho vieme usúdiť, či je splnená podmienka na to, aby nastala z nich vyplývajúca udalosť.

Pri kreslení stromu sa postupuje buď zvislo alebo vodorovne, pričom pri zvislej variante býva vrcholová udalosť vždy na hornej strane a listy, čiže základné udalosti alebo nerozvinuté udalosti, na spodnej strane stromu. Pri vodorovnej variante sa môže strom kresliť z oboch strán. Z hľadiska čitateľnosti pôsobí najpraktickejšie prvá možnosť.

Vďaka malému množstvu grafických symbolov je daná metóda jednoduchá na naučenie a aj dobre čitateľná, keďže sa zakresľuje do stromu. Vďaka tomu, by aj človek, ktorý nemá s touto metódou žiadne skúsenosti mal pochopiť už zostavený strom.

Horšie to v tomto prípade môže byť pri stanovení správnej vrcholovej udalosti. Myslím si, že toto nemusí byť celkom triviálna záležitosť a pri zvolení príliš abstraktnej udalosti sa môžeme dostať do stavu, kedy bude strom príliš košatý a jeho zostrojenie zaberie omnoho viac času. Na druhej strane je stanovenie príliš konkrétnej udalosti, kde zas môže nastať situácia, že nebudeme schopný zostrojiť žiadnu FTA [1,3].

## Aplikácia vo fázach životného cyklu softvéru

Použitie tejto metódy je v životnom cykle softvéru možné na niekoľkých miestach. V ako prvej to je vo fáze identifikácie požiadaviek, kde je možné identifikovať slabé miesta z vytvorenej špecifikácie požiadaviek. To umožní dané požiadavky upraviť, alebo doplniť, čím sa môžme vyhnúť rizikovému stavu, ktorý by mohol ohroziť výsledok vývoja softvérového produktu. Tiež je dobré identifikované nebezpečenstvá sledovať počas

priebehu životného cyklu. Myslím, že v tejto fáze je použitie metódy FTA úplne najužitočnejšie, pretože môže dôjsť k skorému odhaleniu chýb a nejasností, ktoré by sa mohli preniesť do ďalších fáz, kde by skomplikovali situáciu a vyžiadali by si ďalšie zdroje.

Vo fáze návrh sa snažíme identifikovať slabé miesta na vysokej návrhovej úrovni. Cieľom je pomocou vhodných úprav posilniť celkový návrh. Ďalším cieľom je určiť, ktoré časti softvéru majú priamy vplyv na jeho bezpečnosť. Vďaka tomu je neskôr možné sa na tieto časti špeciálne zamerať a tým sa vyhnúť možným neželaným stavom. Aj v tejto fáze má podľa mňa FTA význam a to hlavne v prevencii, keďže zistíme, ktoré časti softvéru by mohli ohrozovať celkovú bezpečnosť, vieme sa na ne vo vývojovej fáze lepšie zamerať.

Použitie tejto metódy je možné aj v realizačnej fáze, kde je však potrebnej oveľa viac práce na vytvorenie stromu poruchových stavov, pretože v tejto fáze sa metóda používa na úrovni kódu, čo je však zdĺhavé a pracovne náročné. Tiež opravy chýb, ktoré sa týmto spôsobom zistia si vyžadujú oveľa viac nákladov ako v skorších fázach. Je však užitočné jej použitie v obmedzenom množstve hlavne na miesta, ktoré sa vo fáze návrhu identifikovali ako rizikové z hľadiska bezpečnosti. Jej hlavným cieľom je identifikovať kritické časti kódu komponentov, ktoré majú priamy vplyv na bezpečnosť softvéru. Ďalším cieľom je pridanie vhodných opatrení, pomocou ktorých sa vyhneme kritickému stavu. Nič to však nemení na tom, že skoršie použitie odhalí mnohé nedostatky už v predchádzajúcich fázach vývoja, čo ušetrí čas aj prostriedky, preto nie je vhodné používať túto metódu iba v realizačnej fáze. Použitie v tejto fáze je skôr na overenie bezpečnosti [2].

## Použitie pri komunikácii so zákazníkom

Pri komunikácií so zákazníkom vo fáze návrhu majú analytici svoje návrhy väčšinou spracované vo forme UML diagramov, ktoré sú však ťažko čitateľné pre koncových používateľov, ktorí nemusia mať skúsenosti s objektovo orientovaným návrhom. UML diagramov, ktoré sa pri takomto návrhu používajú, je niekoľko druhov a bežný zákazník nemusí rozumieť ich obsahu. V tomto prípade sa dá použiť strom poruchových stavov, ktorý je ľahko čitateľný a obsahuje len pár logických pravidiel. Zákazníci dokážu po preskúmaní stromu rýchlo pochopiť jeho princíp a doplniť chýbajúce prvky do komponentov, čím doplnia strom porúch a tým zlepšia kvalitu samotného návrhu, čo ovplyvní aj konečný systém. V tejto oblasti sa pracuje aj na vytvorení transformačných metód medzi UML a FTA, pričom však nie všetky situácie zakreslené v UML je možné transformovať do FTA. Problém je hlavne s časovou súslednosťou a cyklami, ktoré táto metóda momentálne nie je schopná dostatočne zachytiť. Problém súslednosti akcií sa však podaril vyriešiť použitím zložitejších hradiel, otázne však je, či tým metóda nestráca na jednoduchosti, čo je jednou z jej výhod [1,2].

Toto použitie je podľa mňa veľmi zaujímavé a môže sa osvedčiť hlavne so zákazníkom, ktorý nepozná UML. Pri získavaní požiadaviek je takto možné opísať aj negatívne situácie v systéme, o ktorom môže mať zákazník dobré znalosti a tým pádom získame okrem informácie ako systém fungovať má aj informáciu ako fungovať nemá.

## Záver

Metóda stromu poruchových stavov má význam pre použitie v softvérových projektoch, čím by sa malo predísť množstvu nástrah, ktoré pri realizácii projektu číhajú. Jej univerzálnosť ju umožňuje použiť hneď v niekoľkých fázach životného cyklu od získavania požiadaviek cez návrh až po vývoj. Pre čo najlepšie výsledky sa samozrejme odporúča jej použitie čo možno najskôr. Jej jednoduchosť je výhodou, či už vďaka krátkemu času, ktorý je potrebný na jej osvojenie, alebo schopnosti pomocou nej komunikovať so zákazníkmi. Na druhej strane je potrebné pri vytváraní stromu porúch dodržať isté pravidlá, aby mala analýza význam. Výskumy v tejto oblasti naznačujú, že niektoré prvky UML diagramov je možné automaticky konvertovať do FTA, čo môže byť tiež užitočné. Hlavný problém je však so zachytením časových súsledností akcií ako aj cyklov, na ktoré si treba dať pri vytváraní pozor. Určite je vhodný ďalší výskum v tejto oblasti, ktorý by mohol vyriešiť niektoré slabšie stránky tejto metódy, pri čo možno najmenšej strate jednoduchosti a čitateľnosti samotných stromov. Vhodný by bol tiež ďalší výskum konverzie medzi UML a FTA a tiež vytvorenie kvalitného nástroja na tvorbu takýchto analýz, ktoré vďaka možným chybám dokážu eliminovať riziká.

## Použitá literatúra


1. Helmer, G.; Wong,J.; Slagel, M.; Honavar, V.; Miller, L.; Wang, Y.;, „Software Fault Tree and Colored Petri Net Based Specification,“ *International Journal of Information and Computer Security 2007 - Vol. 1, No.1/2 pp. 109 - 142*
2. Massood Towhidnejad; Wallace, D.R.; Gallo, A.M., Jr.;, "Validation of object oriented software design with fault tree analysis," *Software Engineering Workshop, 2003. Proceedings. 28th Annual NASA Goddard* , vol., no., pp. 209- 215, 3-4 Dec. 2003
3. Štetinová, A.: Analýza stromu poruchových stavov – metóda FTA a jej aplikovanie v oblasti cestnej nákladnej dopravy, Žilinská univerzita v Žiline, Fakulta prevádzky a ekonomiky dopravy a spojov, 2006.
4. Zheng Yanyan; Xu Renzuo;, "A Human Factors fault tree analysis method for software engineering," *Industrial Engineering and Engineering Management, 2008. IEEM 2008. IEEE International Conference on* , vol., no., pp.1971-1975, 8-11 Dec. 2008


## Annotation

*Elimination of risks with possible faults*

*Quality and security of software are very important goals in its creation process. This is also a need from customers, who demand reliable and safe systems. Fault tree analysis method is one of the methods used for searching risk points in complex systems and for increase of security. This essay talks about the use of this method in variable phases of the life cycle of software, about forming and progress of this method, about the creation technique and parts which construct the fault three. This method is also applicable for communication with the customer, where it has some benefits, which is also addressed in this essay.*