1 Syfte

Att identifiera orsaker (*hazard causes*) till en oönskad händelse (riskkälla, vådahändelse) och skatta dess sannolikhet (sh), för att bedöma var risklindrande designändringar är nödvändiga eller för att verifiera att systemsäkerheten är tillgodosedd.

2 Skede

Från produktdefinitionsfasen (detaljerad system- / komponentdesign) fram till färdigt system.

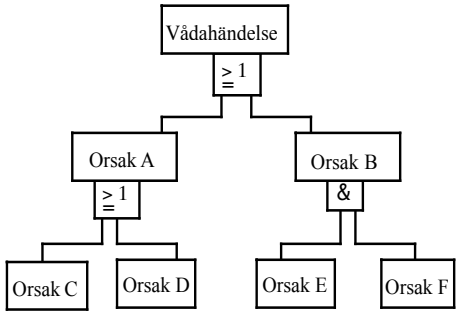
3 Ingångsmaterial

- Detaljerade designbeskrivningar över aktuellt system[1]
- Riskkällelista från tidigare systemsäkerhetsanalyser (t ex HAZOP, FMECA).

4 Resultat

- Ett logiskt felträd (alternativt *truth table*) per vådahändelse och systemtillstånd/systemkonfiguration.
- Förslag på riskreducerande systemändringar[2,14]

Exempel på ett FTA-träd:



5 För- och nackdelar

\+ Systematisk, lättolkad, verktygsstödd grafisk metod för analys av enskilda/multipla orsaker till oönskad händelse.
\+ Kan inriktas mot faktorer och oönskade händelser som berör systemsäkerhet (likaväl som tillförlitlighet)[2].
\+ Effektivast för system med diskreta händelser (till skillnad mot dynamiskt föränderliga system).
\+ Samverkan mellan olika systemdelar[3] samt externa händelser kan modelleras och specifikationsfel detekteras.
\+ Vådahändelsens sannolikhet kan härledas, givet att sannolikheterna för de bidragande faktorerna kan skattas[4].
\+ SFTA, Software FTA, är en specialvariant att applicera på lövnoder som beskriver programvarans logik[5].
\+ Indikerar för programvara vilka riskfyllda villkor att undvika/kontrollera, när under exekveringen och var i koden.
+_ Förutsätter fastställd systemdesign och full kunskap om systemets beteende i alla dess moder.
+_ Ger en ögonblicksbild av systemets tillstånd vid en viss tidpunkt.
_ Tidskrävande/omfattande: Stort antal felträd för varje system (ett per topphändelse och systemtillstånd).
_ Komplexa, svåröverskådliga felträd t ex för *interrupt*-drivna realtidssystem med flera parallella processer.
_ Slutförd analys av enskild nod på viss trädnivå nödvändig innan fortsättning på underliggande nivå.
_ Tidsföljd, tidsfördröjningar, tillståndsövergångar avspeglas ej, partiella och multipla fel svårmodellerade.
_ Design med detaljerad programvarulogik (eller färdig kod) nödvändig vid applicering av FTA på programvara.
_ Endast semi-automatisk trädgenerering från promgramvarukod (loopar komplicerar).

---

[1] Strukturella samt funktionella systembeskrivningar (flödesscheman, interaktionsdiagram, logiska diagram), systemomgivning, systemanvändning etc.

[2] Att jämför med t ex FMEA och ETA, som enbart ser till felyttringar (och därmed tillförlitlighetsaspekten).
Ur identifierade olycksscenarier kan förslag till riskkällereducering hämtas (t ex eliminering av enkelfel, felbeteende med gemensam felorsak/*common cause failure*, händelser som ingår i flera hindermängder).

[3] Ex: Subsystem, komponenter, maskin-/mjukvara, mänskliga faktorn.

[4] Ex på sh:sformler (förutsatt oberoende händelser): en ’och’-grind svarar mot en produkt av sh:er (resulterande sh fjärmas från 1), en ’eller’-grind mot en summering (resulterande sh förs närmare 1). Andra mer komplicerade formler gäller vid beroende händelser. Se fotnot 359 i [5] samt [6]:sid 190,202.

[5] Felträdstemplat finns för olika språkelement (tilldelning, if, while, proceduranrop etc).

_ FTA:s sannolikhetsanalys ej meningsfull på programvarukod (bättre åtgärda identifierade brister i logiken).
_ Kompletterande analysmetoder nödvändiga (t ex betr tidsaspekter).

6 Aktivitet/metod

- Initiera FTA:
  - o Klargör analysförutsättningarna[6].
  - o Upprätta en arbets- o mötesplan för FTA-analyserna[7].
  - o Sätt samman en analysgrupp.
  - o Samla in och distribuera ingångsmaterialet (se 3) för förberedande studier i analysgruppen.
- Håll analysmöte:
  - o Presentera analysunderlaget (se 3).
  - o Bestäm den topphändelse, som skall analyseras
  - o För varje topphändelse:
    - – Definiera systemets fysiska gränser
    - – Bestäm det systemtillstånd för vilket topphändelsen skall analyseras
    - – Fastlägg den aktuella analysens omfattning (stoppkriterier, minsta detaljeringsnivå)[8]
    - – Upprätta ett logiskt felträd med vådahändelsen som topp-nod
    - – Utgå från denna händelse och spåra dess orsaker[9] bakåt i tiden (deduktiv analys):
      - _ Identifiera närmast föregående orsaker (händelse/felkälla eller kombination av händelser)
      - _ Beskriv dessa grafiskt[10] på nästa undernivå i trädet samt den logiska relation[11] som föreligger:
        - o Sammanbind den överliggande händelsen med underliggande orsaker[9] via en:
        - o ’och’-grind, där oberoende, samverkande orsaker föreligger
        - o ’eller’-grind där flera alternativa orsaker är möjliga (vart och ett tillräckligt för att ensamt trigga överliggande händelse).
      - _ För varje händelse på denna nivå i trädet: beskriv närmaste undernivå i analogi med ovanstående.
      - _ Avbryt processen där a) analysen nått ett löv (bas- eller primärhändelse)[12], eller där
        b) nodens riskbidrag bedöms försumbart, alternativt där
        c) en designändring kan minska riskbidraget (vilket gör fortsatt nedbrytning meningslös)[8].
    - – Inled den kvalitativa analysen: Reducera trädet till dess minimala hindermängd[13].
    - – Genomför en kvantitativ analys: Utnyttja de minimala hindermängderna för skattning av vådahändelsens sannolikhet ur bashändelsernas (givet att meningsfulla data finns att tillgå)[4].
    - – Ge förslag på systemförändringar som eliminerar/kringgår vådahändelsen[14].
    - – Stäm av resultaten med systemdesignern
  - o Fortsätt med nästa systemtillstånd för samma topphändelse (dvs nästa felträd).
  - o Övergå till nästa topphändelse (samt de systemtillstånd som där är aktuella).
  - o Dokumentera analysresultaten (se 4):
    - – Notera utestående frågor o oklarheter för vidare utredning av utpekad ansvarig efter mötet.
    - – Slutsignera FTA-mötets resultat och notera gruppkonsensus.

---

[6] Systemets uppgifter/användning, omgivning/begränsningar. Analysens syfte (fokus tillförlitlighet/systemsäkerhet). Enbart kvalitativ analys eller även kvantitativ analys.

[7] Arbetsprocess, startobjekt (vilken vådahändelse som först bör analyseras), generella stoppkriteria för denna typ av analys, tidplan, mallar, begreppsförklaringar. Analysordning: i vilken ordning vådahändelserna bör analyseras.

[8] Analysen kan stoppas t ex om en bidragande orsak kan elimineras (eller dess sannolikhet reduceras) genom en designändring (jfr fotnot 14). Om analysen drivs ned i programvarustrukturen, avgör hur långt (komponentnivå, flödesdiagram etc).

[9] Startvillkor/utlösande händelse, mellanliggande avsedda händelser, möjliga men otillåtna händelser etc, där faktorer såväl i analyserad systemdel som i omgivning, COTS, underliggande resp extern maskinvara, operatörsinteraktion beaktas.

[10] T ex bashändelse (cirkel), mellanliggande pseudohändelse (rektangel), *Transfer* (triangel).

[11] T ex *AND, OR, Priority AND, Exlclusive OR, Inhibit* (6-hörning)*, M of N.*

[12] En händelse som saknar underliggande orsaker (*the root cause of the hazard*).

[13] Den kombination bashändelser nödvändig och tillräcklig för att utlösa topphändelsen (mellanliggande pseudohändelser utrensade). Datorstödd reduktion nödvändig: ett medelstort träd kan innehålla miljontals minimala hindermängder (*minimal cut sets*).

[14] I första hand ändras den del av designen som representeras av ’eller’-grindar (se fotnot 4). Detta gäller i synnerhet enkelfel: väg genom trädet enbart bestående av ’eller’-noder, en systemlösning som inte är tolerabel. Om möjligt bör ’eller’-grindar ersättas av ’och’-grindar, t ex genom att införa ytterligare utlösande villkor (eller flera steg) före oönskad (del)händelse eller införa diversitet. Ex på ytterligare angreppssätt: övergång till en annan realiseringsteknik, införande av skyddsmekanismer och säkerhetsprocedurer.

- Följ upp vidtagna utredningar/åtgärder i efterföljande möte:
  - Identifiera eventuella kvarstående/obesvarade åtgärder/frågor.
  - Ompröva analysresultaten vid förändringar i system–omgivning–användning.

7 Referenser

Förutom de referenser som kan erhållas ur sökningar efter 'FTA/Felträdsanalys/Fault tree analysis' på nätet:

[1] Uppdragsspecifkation SäkAnalysMetoder, FMVdokument 14910:88774/2005.
[2] PHL-analys, Faktablad för Preliminär riskkällelista, FMVdokument 14910:2662/2006.
[3] PHA, Faktablad för Preliminär riskkälleanalys, FMVdokument 14910:2795/2006.
[4] Försvarsmaktens handbok för Systemsäkerhet, M7740-784851 H SystSäk, s 102 ff.
[5] Försvarsmaktens handbok för programvara i säkerhetskritiska tillämpningar, M7762-000531 H ProgSäk, s 176.
[6] Hazard Analysis Techniques for System Safety, C.A. Ericsson, ISBN 0-471-72019-4, s 183 ff.
[7] System Safety Analysis Handbook, www.system-safety.org/ProductsSale.php, s 3-125 ff
[8] Safeware, System Safety and Computers, N G Leveson, ISBN 0-201-11972-2, s 317 ff + 497 ff.
[9] Safety-Critical Computer Systems, N Storey, ISBN 0-201-42787-7, s 43 ff.
[10] Fault Tree Analysis (FTA), IEC 1025.
[11] Retrenchment and the Generation of Fault Trees for Static, Dynamic and Cyclic Systems, Banach et al, Safecomp 2006.

8 Dokumenthistorik

Version	Datum	Beskrivning
1.0	06-09-24	Faktablad enl [1].
1.1	06-10-06	Uppdatering av fotnot 8. Tillägg efter referens till fotnot 12. Nytt: ref [11].