

Fault Tree Analysis By Design

Clifton A. Ericson II; The Boeing Company; Seattle, Washington

Keywords: fault tree, fault tree analysis

ABSTRACT

Fault Tree Analysis (FTA) is a root-cause tool for analyzing and evaluating failure paths in a system, providing a mechanism for system level risk evaluations. Since the inception of FTA, fault tree mathematical methods and computer codes have received greater attention and have been significantly improved. Fault tree (FT) computation has moved from mainframe computers to desktop computers and fault tree plotting has transitioned from specialized mainframe plotters to common desktop laser and inkjet printers. However, fault tree construction methods and theories have fallen behind fault tree computerization development. Knowing how to accurately develop and construct a fault tree is probably the most important step in FTA. If the fault tree is not correct, all of the computerized fault tree tools are essentially meaningless. Most books and papers on FTA only have a cursory section on fault tree construction, which generally describe just the basic fault tree definitions. This paper expands the basic information on fault tree construction, collected from years of experience in building fault trees for various systems.

FAULT TREE CONSTRUCTION

There are various steps, stages and aspects to Fault Tree Analysis, such as construction, cut set generation, cut set evaluation, importance measures, plots, reports and documentation. One of the most important steps, and the least understood and discussed step, is that of the actual fault tree construction. The following are basic, but key rules that are crucial in fault tree construction. Each of these precepts discusses different concepts that should help the analyst construct better fault trees.

Rule #1 – Know The Purpose And Strengths Of FTA.

The very first thing to determine is if FTA is the correct tool for your particular problem. This involves understanding the purposes and strengths of FTA. It is important to select the

right tool, and to correctly apply it to the problem.

Remember, FTA is not a hazard analysis. The purpose of a FTA is not to identify potential system hazards. The purpose is to find out how a given identified potential hazard (Undesired Event) can occur. All of the possible system contributing factors and their relationships are established, and a top probability of occurrence calculated. A FT is a root cause deductive analysis tool that provides a way to logically combine all failures, events and conditions that can lead to the occurrence of an Undesired Event.

Basic considerations include:

- use the right tool
- use the tool correctly
- FTA is a tool for:
 - root cause deductive analysis
 - identifies events contributing to an Undesired Event
 - computes the probability of an Undesired Event
 - measures the relative impact of a design fix
 - fault path diagrams for presentation

Rule #2 -- Know The Purpose And Objectives Of Your FTA Study.

The next step is to establish the goals and objectives of your FTA, and determine if these goals can be suitably met via FTA. This involves understanding the problem, the requirements and the end objectives. Before doing any analysis it is appropriate to determine if the results from a FTA will satisfy the problem objectives. Sometimes FTA is contractually specified. When not specified, the analyst must determine if the use of FTA will suitably resolve the problem.

Basic considerations include:

- solve the right problem/ do the right analysis
- establish a problem/solution statement
 - what is the problem statement

- what are the solution requirements
- show how FTA results will satisfy or solve the problem
- test potential FTA results against the problem
- make sure top Undesired Event (UE) is correct and reasonable
 - correct model
 - reasonable model
 - don't solve the wrong problem
 - don't try the impossible
 - make sure analysis will meet desired objectives/goals

Rule #3 -- Establish Your FTA Ground Rules.

It is very important to quickly establish your FTA boundaries and ground rules. Much time and effort can be wasted with too much analysis time spent in unproductive or unnecessary areas, on the wrong level of detail or even on the wrong problem.

At the same time, it is important to establish the scope of the analysis with the customer and obtain concurrence. The intent is to establish a FTA contract, so that everyone involved understands the what, where, when and how of the study. This will help avoid future conflicts, since it is easy for different technical experts to see the analysis differently.

Basic considerations include:

- define and document assumptions
- scope the problem
 - size
 - level of analysis
 - level of detail
- set analysis scope and boundaries
- establish analysis definitions
- make sure top UE is correct and reasonable (do the right analysis)
- publish FTA ground rules before starting (living document)
 - definitions
 - scope
 - boundaries
 - level of detail and analysis depth
 - construction rules
 - FT format
- obtain agreement on ground rules
 - design team
 - customer

Rule #4 -- Design Your Fault Tree.

A well constructed fault tree requires some forethought. Things done in the early phases of the fault tree construction can adversely impact later phases of the analysis. It is often difficult or impossible to make certain needed FT construction changes at a later development stage if the FT is very large, and distributed between different analysts or organizations.

FT design is a balance of objectives, capabilities, available data, tools, time and money. Small Ft's (< 100 event) do not require as much actual design effort. But, medium (100 to 1,000 events) and large FT's (> 1,000 events) do require design and planning.

Basic considerations include:

- FT should be based on actual design data
- follow FTA ground rules and formats
- make checks against ground rules
- establish event name conventions
 - use a methodology
 - by hardware type, supplier, subsystem
 - short names are usually better
 - long names becomes burdensome
 - if long names are used, be sure names are distinct, clear and correct
- establish name methodology for transfers
- establish MOE conventions and tracking
- maintain event databases and cross references
 - basic failure events
 - gate events
 - condition events
 - MOE's
 - transfers
- establish tree structure approach
 - functional
 - subsystem
- determine level of analysis detail
 - subsystem
 - LRU
 - component
- be very descriptive in writing event text
 - avoid using word "fail" -- not enough information ("power supply fails" vs. "power supply does not provide +5 VDC")
 - do not use the terms primary failure or secondary failure (provide more description)

- use gate types cautiously
 - AND, OR and Inhibit gates do almost everything
 - if you think an exotic gate is necessary, that's the first clue to re-analyze your problem
- use FT plotting or drawing programs with a consistent professional output
- maintain tree metrics
 - total Basic Events
 - total Gate Events
 - complexity
- tree size vs. effort
 - small (< 100 event)
 - medium (100 to 1,000 events)
 - large (> 1,000 events)
- conduct tree peer review
 - other FT experts
 - system designers
- the model and design data can be iterative
 - preliminary model progresses to detailed model

Rule #6 -- Understand Your Failure Data.

Understanding the system failure data is also another critical step in fault tree construction. This includes both qualitative data and quantitative data. The age old saying “*garbage in - garbage out*” (GIGO) applies to FT data and FTA. Incorrect or misinterpreted data can result in an incorrect model and erroneous results.

Component failure data refers to failure modes, failure rates and data confidence. Reliable proven failure data is obviously the most desirable for the best results. On the other hand, FTA can still provide very useful information using preliminary data and estimates, as long as the analyst fully understands the process and the results.

Rule #5 -- Know Your System.

It almost goes without saying that in order to analyze a system by fault tree, you must first fully understand the system design and operation. One of the basic attributes and strengths of FTA is that it forces you to really understand the system design and operation, in order to accurately build a fault tree model of that system.

Knowing and understanding the system design and operation for FTA does however include many different and important considerations.

Basic considerations include:

- failure data must be obtainable for quantitative evaluation
- data accuracy and trustworthiness must be known (confidence)
- data estimates are useful and can be used, but must be understood
- must understand failure modes, failure mechanisms and failure rates

Basic considerations include:

- know the system design and operation
- know the interfaces between subsystems
- utilize all sources of design information :
 - drawings
 - procedures
 - block diagrams
 - flow diagrams
 - FMEA's
 - stress analyses
 - failure reports
- drawings and data must be current for current results
- requires system engineering skills -- electronics, mechanics, software, etc.
- make periodic checks to make sure the FT model is correct
 - peer review
 - design team review
 - customer review

Rule #7 -- Know Your Fault Tree Tools.

FT tools come in many different sizes, shapes and colors. That is, every tool has different capabilities, features, idiosyncracies and levels of difficulty. Knowing and understanding your fault tree construction and evaluation tool is very important. Some tools are better than others. Some tools provide features that other tools do not. And, one fault tree analysis may require certain tool features that another FT analysis does not need. For example, one project may require only a single phase analysis, whereas another project may require multiple phases with repair.

Basic considerations include:

- know basic tool capabilities
 - construction
 - editing

- plotting
- reports
- cut set evaluation
- know tool user friendliness
 - intuitive operation
 - easy to use and remember
 - changes are easy
- single vs. multi-phase tree
- qualitative vs. quantitative evaluation
- simulation vs. analytical evaluation (considerations include size, accuracy, phasing)
- know tool limits
 - tree size
 - cut set size
 - plot size
- understand cutoff methods, some can cause errors
- gate probabilities are incorrect when MOE's are involved

Rule #8 -- Understand (Appreciate) Small Numbers.

When fault trees are quantified, it is very important to have a clear grasp of small numbers and their meaning. Sometimes it is difficult to truly understand small numbers, but having an appreciation for them is very beneficial. Since we don't use small numbers everyday, people often erroneously deal with small numbers, without fully grasping their meaning or relevance.

Fault trees generally involve rare events, or events with a large Mean Time Between Failure (MTBF). This translates into small failure rates or small numbers (eg, 1.0×10^{-6} , 1.0×10^{-15}). Knowing how to work with these numbers is crucial.

Knowing how to work with small numbers is crucial to FT construction and analysis. Figure 1 is an attempt to place some perspective on small numbers. Probabilities range between 0 and 1, and small numbers are very close to the 0 end of the range. It often makes more sense to compare the relative range of small numbers.

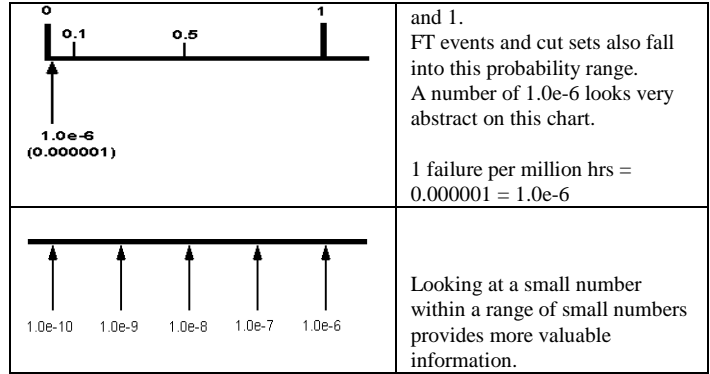


Figure 1 -- What Are Small Numbers ?

Basic considerations include:

- failure rates and probabilities are between 0 and 1
- FT's generally deal with small numbers (< 1.0×10^{-6})
- small numbers are somewhat abstract (see Figure 1)
- the exponent size is of prime interest (e-6, e-15, e-35)
- decimal places are somewhat significant within the same range (1.11×10^{-6} vs 1.97×10^{-6})
- decimal places are not as significant for a wide range (1.1×10^{-6} vs. 1.778×10^{-9})
- as numbers get very very small, decimal place are probably insignificant (ie, 1.0×10^{-35})
- all results are essentially estimates for relative comparisons
 - is system 1.0×10^{-3} or 1.0×10^{-7} is relevant
 - is system 1.1×10^{-6} or 8.7×10^{-6} is not as relevant
 - is system 1.1×10^{-6} or 1.123767×10^{-6} is not relevant
- don't get carried away with numbers

Rule #9 -- Understand Your Results.

Results obtained from a FTA must be well understood, especially before they are released with conclusions and recommendations. It is very easy to misinterpret FT results, misuse FT results or fail to recognize an error in the FT or the computerized results.

Evaluate your FTA results carefully, take nothing for granted, especially nothing from a computer. There are many places for errors to occur, such as computer errors, data errors, use of outdated drawings, logic errors, etc.

| | |
|--|--------------------------------|
| | Probability range is between 0 |
|--|--------------------------------|

Probably the most important item in this step is to make reasonableness tests on all results. That is, make hand checks on the results and conclusions, particularly if they are derived from a computer program. Check all significant CS's to determine if they are valid and reasonable. Calculate by hand the top probability from the most relevant CS's, and compare against the computer results for reasonableness. If a CS or probability is not valid or reasonable, the FT must be checked for errors in logic. If none can be found, then the FT computer program must be scrutinized and tested for accuracy.

Basic considerations include:

- make reasonableness tests on the results
 - take nothing for granted from the computer
 - are CS's credible and relevant, if not revise tree
 - test your results via hand calculations
- effect of MOEs is very important
 - they can cause large numerical impact or none at all
 - handle carefully
- probability calculations are important, but nothing more than a mathematical exercise
- CS's are very important -- shows where to fix system, importance of specific events
- if exotic gates are used, check results, check assumptions
- review step #2 on data and its impact
- verify that the FTA goals were achieved, was the right tool used
- look for analysis errors
 - data
 - model
 - computer program

Rule #10 -- Publish/Document Your Analysis And Results Completely.

Following completion of the FTA it is general practice to document the results. If documentation is not required, it is good practice to do so anyway. The documentation should not just summarize the FTA results. Documentation should include the ground rules, definitions, design information, FT diagrams, FT metrics, quantitative results and conclusions. Complete documentation will help provide a record of the entire process, so that at some future time, analysts can go back and easily remember how

the analysis was conducted. This is useful for future changes or updates to the FTA (sometimes years later).

Basic considerations include:

- complete documentation
 - problem statement
 - definitions
 - ground rules
 - references
 - relevant design data
 - data and sources
 - FT diagrams
 - tree metrics
 - tool description
 - results
 - conclusions

CONCLUSION

Fault tree construction is an important phase in the actual FTA process. Much time and thought must go into fault tree analysis, development and construction. A good fault tree is actually designed, just as a system is designed. This is an area often overlooked by books and technical papers. Several key elements of fault tree construction have been discussed, which should help an analyst build better fault trees. Fault trees are an important tool in risk analysis. FT construction methods, techniques and tools require as much attention and research as FT mathematics. The step into FT Synthesis (automatic FT construction) will require a good understanding of FT construction methods and theories.

REFERENCES

- [1] A. B. Mearns, Fault Tree Analysis : The Study Of Unlikely Events In Complex Systems, Boeing/UW System Safety Symposim, 1965.
- [2] D. F. Haasl, Advanced Concepts In Fault Tree Analysis, Boeing/UW System Safety Symposim, 1965.
- [3] P. M. Nagel, Importance Sampling In System Simulation, Annals Of R & M Symposium, 1966, p330-337.
- [4] C. A. Ericson II, System Safety Analytical Technology -- Fault Tree Analysis, Boeing document D2-113072-2, 1970.

- [5] R. E. Barlow & J. B. Fussell & N. D. Singpurwalla, Reliability And Fault Tree Analysis, Conference On Reliability And Fault Tree Analysis; UC Berkeley; SIAM Pub, 1975.
- [6] J. B. Fussell, Fault Tree Analysis - Concepts And Techniques, Generic Techniques In Systems Reliability Assessment, Noordhoff Publishing, 1976, p133-162.
- [7] N. H. Roberts & W. E. Vesely & D. F. Haasl & F. F. Goldberg, Fault Tree Handbook, NUREG-0492, 1981.
- [8] R. E. Altschul & P. M. Nagel, The Efficient Simulation Of Phased Fault Trees, Annuals Of R & M Symposium, 1987, p292-296.
- [9] J. D. Andrews & T. R. Moss, Reliability and Risk Assessment, Longman Scientific & Technical, 1993.
- [10] E. J. Henley & H. Kumamoto, Probabilistic Risk Assessment And Management For Engineers And Scientists, IEEE Press (2nd edition), 1996.
- [11] C. A. Ericson II, FTAB -- A New Generation Fault Tree Code, 15th International System Safety Conference, 1997.

safety at the University of Washington. Mr. Ericson holds a BSEE from the University of Washington and an MBA from Seattle University.

BIOGRAPHY

Clifton A. Ericson II
The Boeing Company
18247 150th Ave SE
Renton, WA 98058 USA
phone 253-657-5245
fax 253-657-2585
email clifton.a.ericson@boeing.com

Mr. Ericson works in system safety on the Boeing 767 AWACS program. He has 33 years experience in system safety and software design with the Boeing Company. He has been involved in all aspects of fault tree development since 1965, including analysis, computation, multi-phase simulation, plotting, documentation, training and programming. He has performed Fault Tree Analysis on Minuteman, SRAM, ALCM, Apollo, Morgantown Personal Rapid Transit, B-1 and 737/757/767 systems. He is the developer of the MPTREE, SAF and FTAB fault tree computer programs. In 1975 he helped start the software safety discipline, and has written papers on software safety and taught software