Fault Tree Analysis as a Means to Promote Safety

Designing safe products and systems is critical to reduce accident risk and minimize product liability. There are various reliability and quality analysis methodologies used to ensure safety including risk analysis techniques such as Fault Tree Analysis (FTA), Failure Mode and Effects Analysis (FMEA), and others. This article focuses on providing an introductory look at FTA and its role in promoting safety. We review basics about FTA, including what it is, its history, its uses, and advantages. A thorough review of qualitative and quantitative FTA results, including minimal cut sets (MCS), quantitative metrics such as unavailability, and importance measures, is included. The article concludes with general guidance on how FTA qualitative and quantitative results can be used to design inherently safer products and systems.

RELYENCE CORPORATION

Table of Contents

What is Safety Analysis?

Safety is a key element in product design and is a vital aspect of design engineering. Safety is one of the core elements of RAMS programs, or the study of reliability, availability, maintainability and safety in product design and manufacturing. In particular, safety analysis is performed to evaluate ways to prevent harm to people and the environment by a product, system or process. It is used to help eliminate or mitigate overall risk.

Safety analysis can be performed using various methodologies, including:

- Risk analysis – a technique used to identify and analyze potential catastrophic or critical events
- Hazard analysis – an approach that is used to assess risk associated with identified hazards
- Probabilistic risk analysis (PRA)/probabilistic safety analysis (PSA) – a systematic and comprehensive methodology used to evaluate risks associated with a complex engineered technological entity (such as an airliner or a nuclear power plant)

Historical Roots of Safety Analysis

Unfortunately, there are various major historical events that prompted more considerations for safety. In general, there is documentation noting the need for workplace safety starting from the late 1800s through to the 1970s and continuing today. For example, you may see signs at workplaces indicating so many days since the last safety incident.

There were also various major tragedies that sparked more development in the safety analysis area. They include two major disasters in the nuclear power industry: one in 1979 at Three Mile Island in the United States and the second in 1986 at Chernobyl in the then Soviet Union. There was the Bhopal gas tragedy where a pesticide plant in India released a toxic gas that led to many fatalities. Sadly, there were also the two tragic events in the United States space exploration program with the loss of the Challenger and Columbia crews and aircraft in 1986 and 2003, respectively.

The Need for Safety Analysis

Safety analysis has helped to implement and employ techniques designed to prevent tragedies in the future. Current and future technologies suggest the need for continued safety analysis in many areas including aerospace, transportation, manufacturing, healthcare/medical, military, nuclear power, and more. A few examples include:

- In the aerospace industry, safety must be considered to allow for continued safe air travel, safe space exploration, and successful use of UAVs and drones.
- In the transportation realm, there is obvious interest in continued safe travel in both traditional and autonomous vehicles, as well as the need for safe public transportation.

When we design systems and processes with safety in mind, we can:

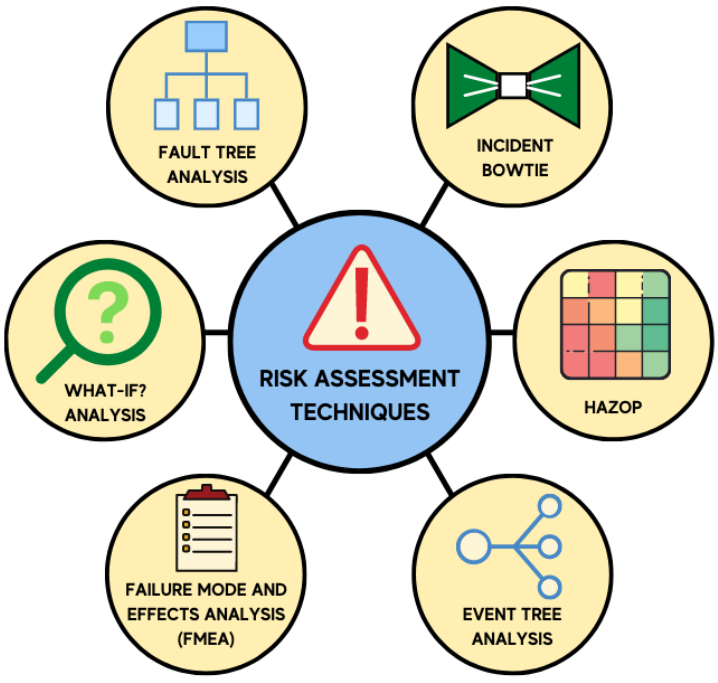
- Avoid accidents that result in injury or death
- Avoid financial and property losses
- Assume risk in a responsible manner

Risk Assessment Techniques Used for Safety Analysis

There are various risk assessment techniques used for safety analysis. They include Fault Tree Analysis (FTA), Failure Mode and Effects Analysis (FMEA) or Failure Mode, Effects and Criticality Analysis (FMECA), Event Tree Analysis (ETA), What-If analysis, Hazard and Operability analysis (HAZOP), Incident BowTie, and others.

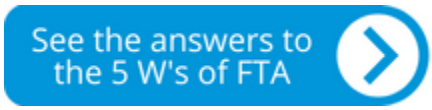
- FTA is a deductive procedure used to determine the various combinations of hardware failures, software failures, human errors, and other events that could cause undesired events (referred to as top events) at the system level.
- FMEA is an organized, systematic approach for assessing potential system failures and the resulting consequences of those failures. The objective of a FMEA is to evaluate the risk associated with the identified failure effects and come up with a plan to detect, prevent, or mitigate those deemed most critical.

- ETA identifies possible outcomes following an initiating event, and the success/failure of subsequent events.
- What-If Analysis is used to identify hazards, hazardous situations, or specific event sequences that could produce undesirable consequences.
- HAZOP, Hazard and Operability Analysis, is a structured and systematic technique for system examination and risk management. In particular, HAZOP is often used as a technique for identifying potential hazards in a system and identifying operability problems that are likely to lead to nonconforming products. HAZOP is based on a theory that assumes risk events are caused by deviations from design or operating intentions.
- Incident BowTie analysis combines the advantages of two other analysis methods: BowTie risk analysis and Tripod incident analysis. The information from BowTie analysis is used as the input for Tripod incident analysis. Consequently, this allows a broader perspective view and makes sure all the possible scenarios are taken into account. By using real-life data, the input from the Tripod incident analysis can be used to make the BowTie analysis more realistic and up to date.



What is Fault Tree Analysis?

Fault Tree Analysis is a top-down, deductive approach to failure analysis. You start by identifying an undesired event - generally some failure or malfunction. Of course, when looking from a safety perspective, that undesired event is generally thought to be "catastrophic." Examples include events such as "aircraft crash with loss of life," "inadvertent arming of a warhead," "vehicle catches fire," "coffee maker catches fire," or "drone crash or collision."

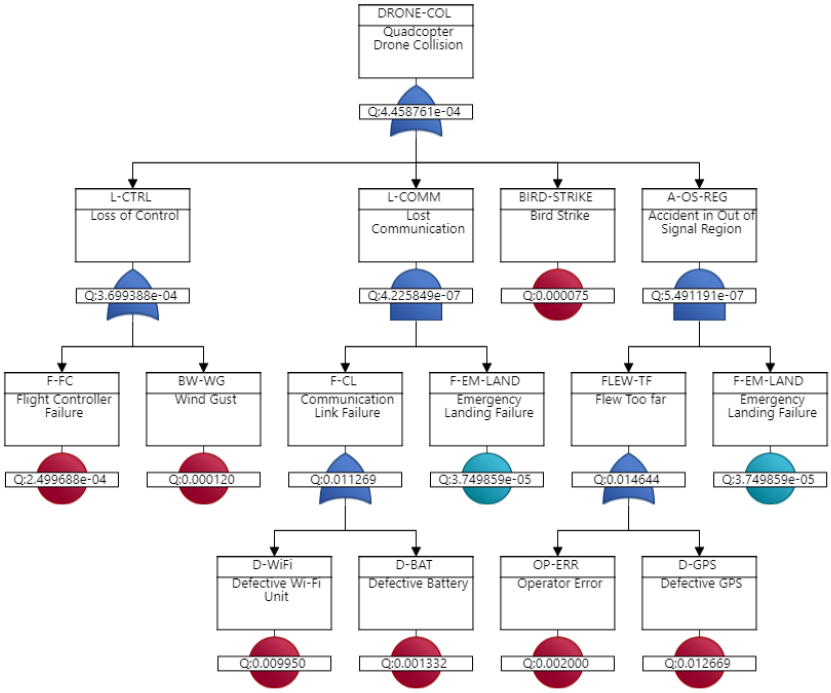


Once you have that undesired event defined, you can use intermediate events, logic symbols and terminal events to model all the ways the undesired event could happen. Note that:

- Intermediate events are events that are developed further in the fault tree using logic symbols.
- Logic symbols are used to construct the fault tree, defining what events alone or in combination lead to any intermediate event or the top undesired event.
- Terminal events are primary events and end each branch of a completed fault tree.

In the end, you have a compact, graphical, intuitive method to analyze a system for safety considerations and overall reliability and availability.

Since the focus here is on safety analysis, examples listed earlier focused on catastrophic failures, but the top event can also be something "unfortunate" but not catastrophic, such as "car doesn't start," "washing machine overflows," "no light on in a room on demand," or "doorbell fails to ring."



Are You New to Fault Tree Analysis?

If you are new to Fault Tree analysis and have not yet implemented a structured fault tree analysis program, you can review our blog for a number of informative fault tree related articles. You can also learn how you can benefit from using a Fault Tree Analysis software tool to manage your analysis.

New to FTA? Learn more

History of Fault Tree Analysis

Fault tree analysis was developed in 1962 by Bell Telephone Laboratories for the United States Air Force for use with the Minuteman system – the weapon system on alert during the Cuban missile crisis. Shortly thereafter, fault tree analysis was used extensively by the Boeing Company with commercial aircraft. In general, fault tree analysis is widely used to investigate the reliability and safety of various systems, often those which are large and complex. Due to concerns for human safety, fault tree analysis has been applied extensively in the nuclear and aerospace industries as well.

Uses of Fault Tree Analysis

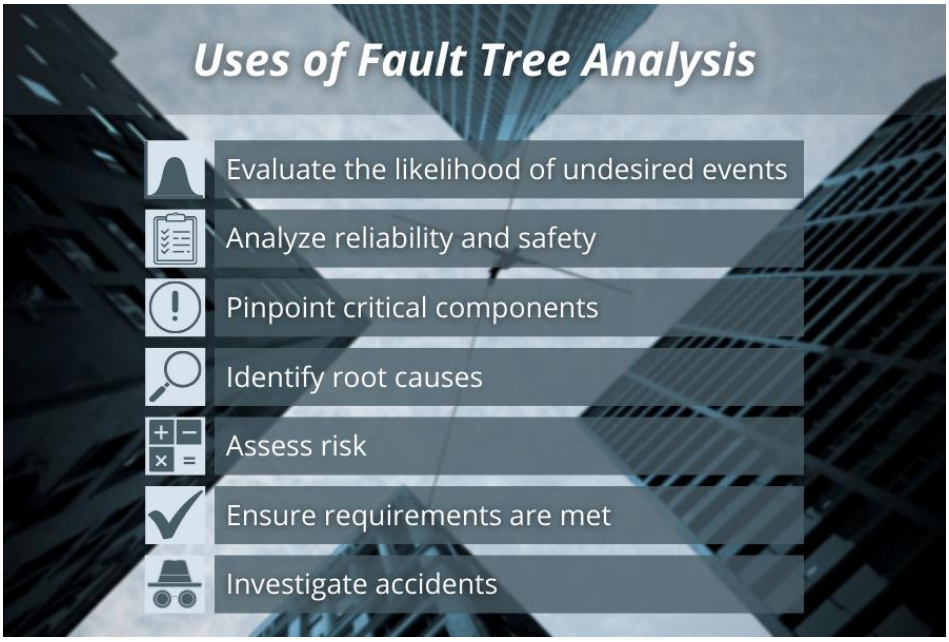
First and foremost, fault tree analysis enables us to evaluate the probability of occurrence of undesired events. If these reliability or safety issues can be identified during the design phase, we can prevent potential catastrophic situations by making changes to eliminate or mitigate those issues.

Fault tree analysis can also be used to:

- analyze reliability and safety analysis during operation.
- identify components that might need further scrutiny or that are at the root of possible safety issues.
- identify root causes of possible failures.
- assess product or process risk.
- help certify that requirements are met.
- aid with the investigation of accidents or incidents.

More specifically, here are some examples of uses for fault tree analysis:

- In the aerospace industry, it can be used to help prevent passenger and crew injuries or fatalities as well as injuries or fatalities on the ground.
- In the medical industry, it can be used to find ways to prevent death during an operation or to prevent incorrect medicine or dosage.
- In the nuclear power industry, it can help to prevent accidents associated with complex nuclear power plants.



Benefits of Fault Tree Analysis

There are many advantages to utilizing fault tree analysis, including:

- It is widely known and accepted as a useful technique for reliability and safety analysis.
- It provides a visual display of the failure behavior of a physical system for easier understanding.
- Both qualitative and quantitative analyses are possible for high-level review and detailed in-depth metrics-based analysis.
- It is based on commonly known methodologies such as Boolean algebra and probability theory. Knowing the underlying principles are based on well-established concepts gives confidence to those doing the analysis and reviewing the results.
- You can analyze simple or complex systems in an organized, logical, systematic way.
- It is not limited to considering issues due to hardware and software failures. It allows you to incorporate other possible concerns such as human error and environmental conditions and how such events can contribute to undesired events.
- Because you can consider combinations of events leading to the top undesired event and not just single failures, you can analyze complex systems.

Challenges of Fault Tree Analysis

As with any risk assessment technique or any additional methods you consider as part of your analysis processes, it's worth considering some of the challenges of fault tree analysis.

- You must be forward thinking and be able to anticipate the top-level undesired event as well as contributing events.
- It is important to have analysts who have in-depth knowledge about the system under analysis.
- It is critical to define your top undesired event as concisely as possible. If the definition is too general, you might end up with a fault tree that is difficult to manage and gain significant benefit from. On the other hand, if defined too specifically, it will limit your fault tree and may not lead to useful results.
- It can be a time-consuming exercise. However, you can realize a significant return on your investment by not only improving product and process safety, but also reliability and quality.

Fault Tree Analysis Basics

In order to successfully build a fault tree for fault tree analysis, it is critical to understand Logic and Event symbols, make appropriate preparations for your fault tree construction, and understand rules and conventions for fault tree construction.

Understanding Logic and Event Symbols

To effectively use FTA, you need to understand the basics of the Logic and Event symbols used in fault tree analysis. You will need to use both Logic symbols and Event symbols.

Logic symbols, like Boolean AND and OR gates, can help you to link branches of the fault tree together.

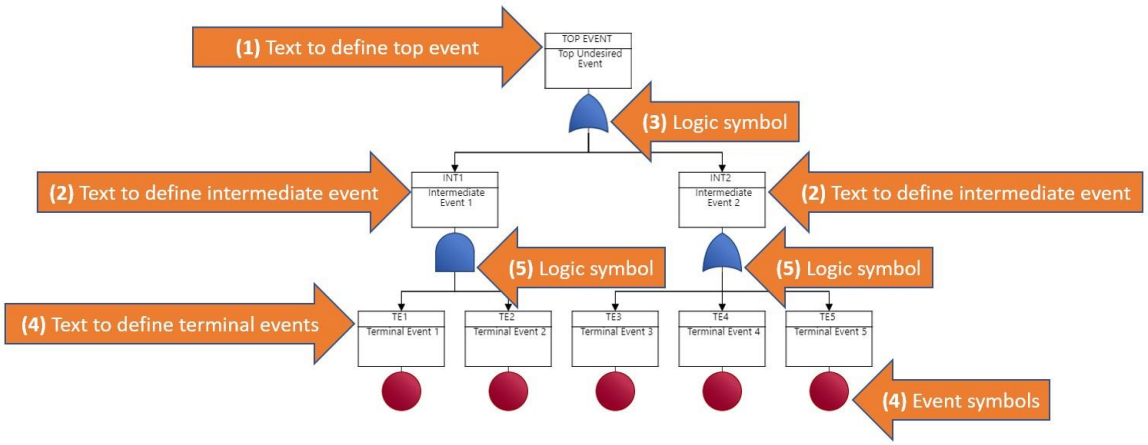
- The AND gate is used to indicate that the output, or the event represented by the AND gate, occurs if and only if all input events occur.
    - In a simple 2-input example, a text message fails to send (the undesired event) if one cannot connect to Wi-Fi (one input event) AND there is no cellular signal (a second input event). So, we'd have two Event symbols connected by the AND Logic symbol.

- The OR gate is used to indicate that the output, or the event represented by the OR gate, occurs if and only if at least one of the input events occur.
    - For example, you might lose control of my drone if you fly too far out of range (one input event) OR a strong wind gust occurs (a second input event). Either of those events or both can cause the event represented by the OR gate (loss of control of the drone) to be true.

Event symbols are used to represent primary or simpler events. They are terminal events in any fault tree branch. Commonly used types are Basic and Undeveloped events.

- Basic events are most commonly used to model hardware failures (i.e., capacitor C12 fails short), software failures, human errors, and other terminal events.
- Undeveloped events are much like Basic events, but often are used to represent events that could be further developed (if time and resources allow and the need arises).
- Examples of basic and/or undeveloped events include:
    - Cannot connect to Wi-Fi
    - No cellular signal
    - Fly too far out of range
    - Strong wind gust occurs

In the generic fault tree pictured here, we have both Logic symbols and Event symbols as well as text to help define the Top, Intermediate and Terminal events.



(1) The text that defines the undesired top event.
(2) The text that defines the contributors to that top event, some intermediate events.
(3) Logic Symbols define how those events, alone or in some combination, impact the next higher event.
(4) Terminal events for each branch (text-based descriptions included), are all defined with their Event symbols.
(5) Logic symbols that define how the terminal events contribute to the intermediate events.

This is just a simple fault tree used for explanation purposes. Fault trees can grow quite large and contain many levels and branches.

And while AND and OR gates are the most commonly used Logic symbols and Basic and Undeveloped events are the most commonly used Event symbols, there are others you can incorporate where needed. Many of the most common examples of Logic symbols and Event symbols are pictured below.

Learn more about Logic gates and Event symbols in our in-depth overview.

Preparing for Fault Tree Construction

It is important to consider what you need to construct a fault tree. Of course, you need a good understanding of the system or process you want to analyze with fault tree analysis. System block diagrams, functional block diagrams, and FMEA data might all be useful in preparation for building a fault tree.

By starting with a clear understanding of the event you want to model with fault tree analysis, you start off on the right foot with a well-defined top event. In addition to understanding your top event, you must be able to define the contributing factors. And, as with many types of analysis, defining any key assumptions can also be helpful.

Finally, you should also think about what metrics are required or what outputs are expected. That will help you identify what key inputs are required. For example, if you only require qualitative analysis with minimal cut set analysis, you don't need to worry about quantitative data for the terminal events. If you do require some quantitative outputs, you need to identify probability of failure, frequency, or other data for the events.

Understanding Rules and Conventions for Fault Tree Construction

It can also be useful to consider the following general rules and conventions that can be considered when building and analyzing a fault tree:

- Identify the level of detail required. As you build out the fault tree, by knowing the level of detail required, you'll know when a branch can end with a terminal undeveloped event versus when a branch must be developed further to terminal basic events. For example:
    - One of the deciding factors for the level of detail is to develop to the level of detail where you have control. Perhaps you have a subsystem delivered to you by a third-party vendor so you don't know all the details and events that could lead to failures related to that subsystem; in that case, you might opt to end relevant branches with undeveloped events or perhaps even require that third-party vendor to supply a fault tree and relevant data.
- Use consistent naming conventions for Logic symbols and Event symbols. You want to describe both what can fail and how it can fail.

- Remember to think beyond just component failures. In addition to component or hardware failures, also consider things like software failures, human errors, and other outside influences.
- Involve a cross-functional team in order to get the most thorough analysis possible.

How to Perform FTA: A Step-by-Step Example

Once you understand the basics of fault tree analysis, including Logic and Event symbols, key rules, and conventions, and have prepared for fault tree construction, you can perform the following steps to construct a fault tree and complete the fault tree analysis:

1. Build the fault tree
   - Define the top-level failure or event
   - Define the contributing factors to the top-level failure
   - Develop the contributing factors to the top-level failure
2. Perform qualitative and/or quantitative analysis
   - Evaluate Cut Sets
   - Calculate Metrics
   - Analyze Importance Measures
3. Take steps to improve the safety of your product or process

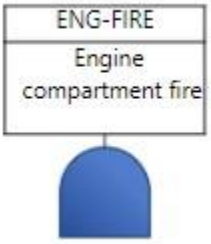
Let's take a detailed look into each of these steps using an example case of a vehicle engine compartment fire. The first step centers around the creation of the fault tree itself. The last two steps focus on the analysis portion of FTA.

Step 1: Build the Fault Tree

The first step in FTA is to create the fault tree diagram to be used for analysis. As previously mentioned, it is critical to be methodical and consider the scope and detail that will provide the most meaningful results for you.

Define the Top-Level Failure or Event

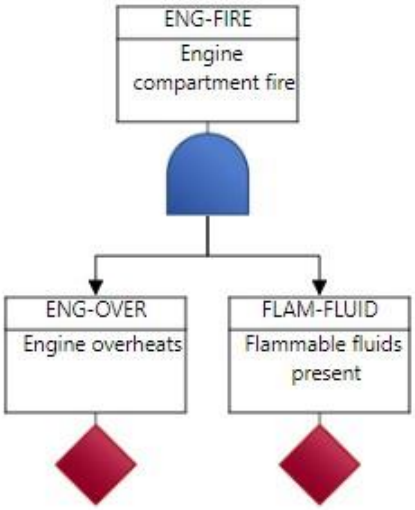
For our example vehicle engine compartment fire, you could start with something like this:



It includes a top event with a description and the initial Logic symbol as an AND gate, with the expectation that several lower-level events together would lead to the vehicle engine compartment fire. You could, of course, adjust that Logic symbol from the AND gate to something else as needed as you develop the fault tree.

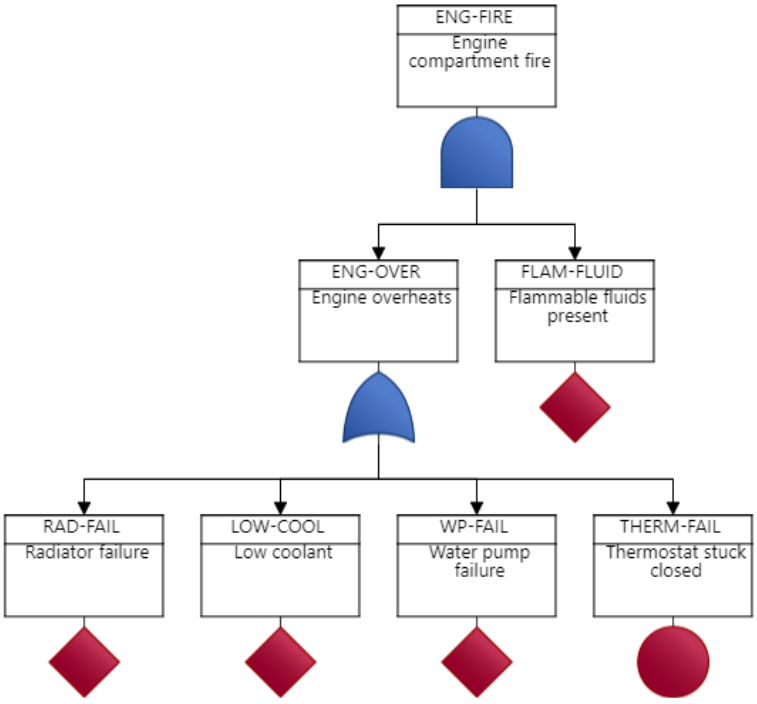
Define Contributing Factors to the Top-Level Failure

Next, you define the first-level contributors. In the example, assume the engine compartment occurs if the engine overheats and there are flammable fluids present. Initially, you could simply provide descriptions of these contributors and define them as undeveloped events.

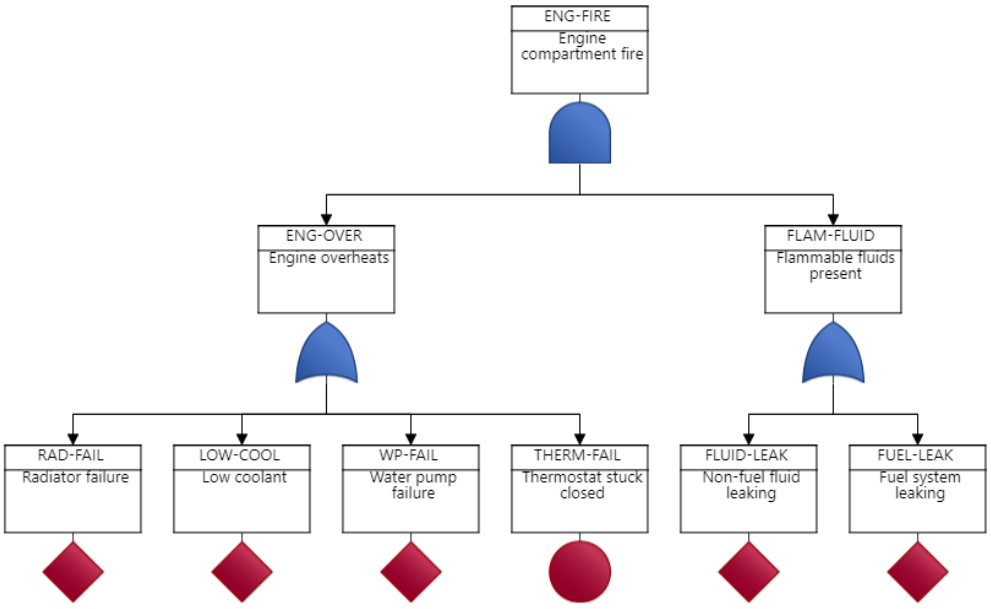


Develop the Contributing Factors

In the example fault tree, you might first develop the fault tree further as it applies to the possibility of the engine overheating. The undeveloped event representing the engine overheats contributing factor can be changed from the initial undeveloped event to a Logic symbol, such as an OR gate. Then the next level events can be defined as shown below.

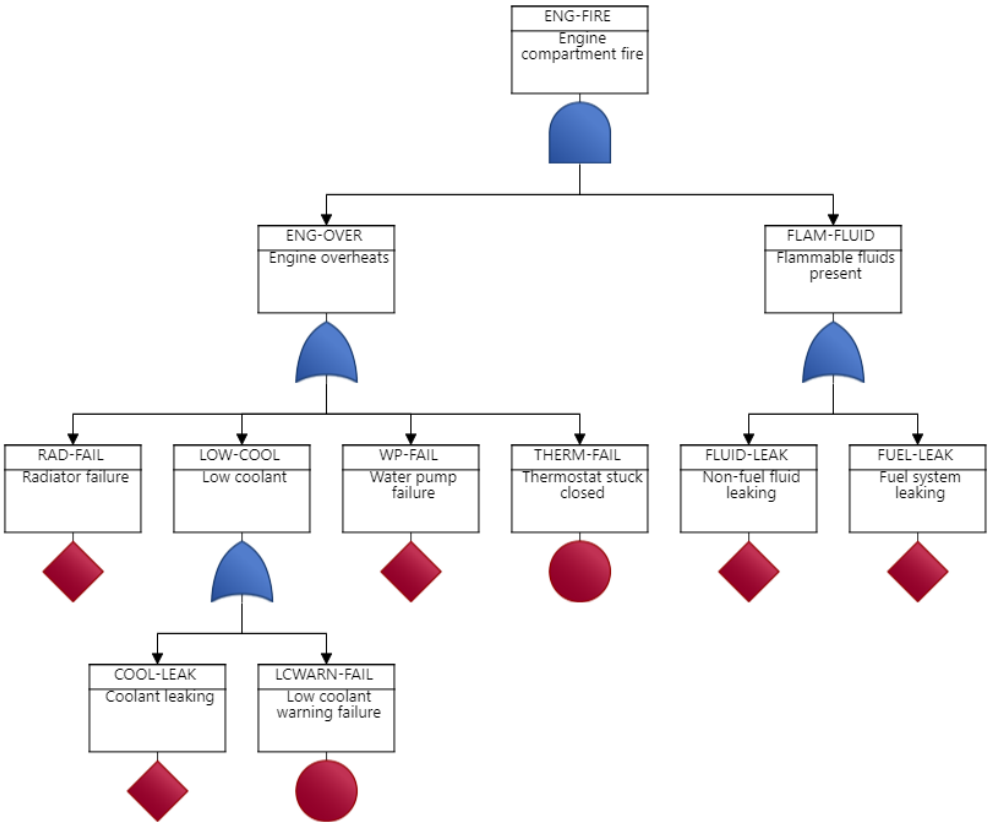


And next, you could develop the tree further as it applies to the possibility of the flammable fluids are present. Switch the flammable fluids present undeveloped event to an OR gate and define the next-level events; for example, 2 undeveloped events to cover non-fuel fluid leaking and fuel system leaking.



Finally for the example, assume some terminal events will be left as-is, but the low coolant undeveloped event will be further developed. Switch the low coolant undeveloped event to a Logic symbol such as an OR gate and define its contributors such as coolant leaking and low coolant warning failure.

With all branches ending in terminal events, assume this is the level of detail to which you plan to develop this fault tree. If there is interest or a requirement and time allows, any of the terminal undeveloped events could be further developed.



Step 2: Perform Qualitative and/or Quantitative Fault Tree Analysis

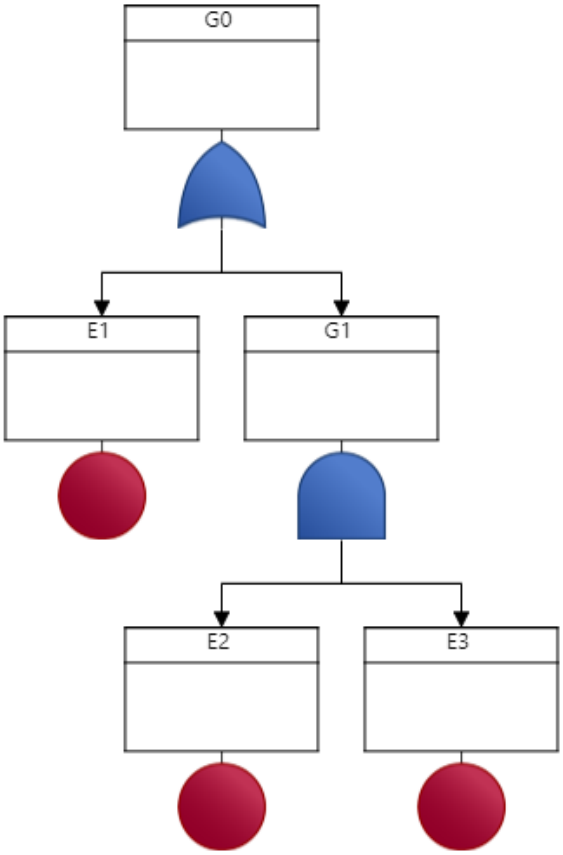
After building our fault tree, the next step is to analyze the diagram in order understand the risk and safety profile our system. To do this this, qualitative and/or quantitative analysis is utilized. From a qualitative point of view, we perform *cut set analysis*. From a quantitative point of view, we calculate various *metrics* and *importance measures*. Additionally, if we have quantitative data for our terminal events, we can also determine the quantitative metrics of the various cut sets.

Evaluate Cut Sets

A cut set is a collection of fault tree terminal events, such that if they all occur, the top event will occur. Note that most analysts are interested in ***minimal*** cut sets, which is the smallest collection of fault tree terminal events, such that if they all occur, cause the top event to occur. The use of minimal cut sets helps highlight issues in a more focused way. Cut set analysis is a thorough and systematic way to find minimal cut sets.

A Simple Cut Set Example

The following simple example fault tree can help to illustrate cut set and minimal cut sets.



There are three cut sets for the undesired top event represented by G0. They include:

- E1
- E2 – E3
- E1 – E2 – E3

That means that the event represented by G0 occurs if E1 occurs, if E2 and E3 occur, or if E1, E2, and E3 all occur. But we can reduce to minimal cut sets by eliminating the E1 – E2 – E3 cut set since it is a superset, or a cut set that includes other cut sets. It includes cut sets E1 and E2 – E3.

And therefore, the minimal cut sets for the undesired top event represented by G0 include only:

- E1
- E2 – E3

Cut set analysis is what is ultimately used to determine minimal cut sets. Several cut set analysis methods include the Inspection method as well as various cut set generation algorithms.

Evaluating Cut Sets using the Inspection Method

Using the Inspection method for identifying cut sets and minimal cut sets, you simply review the fault tree, considering all Logic symbols, and determine the combinations of terminal events which lead to the top event occurrence. The Inspection method requires human intervention and therefore is not applied using algorithms via computers. Thus, the Inspection method is only practical for relatively small fault trees.

Evaluating Cut Sets using MOCUS and MICSUP Algorithms

Cut set algorithms provide a systematic approach to finding minimal cut sets.

Two well-known cut set generation algorithms include:

- MOCUS
- MICSUP

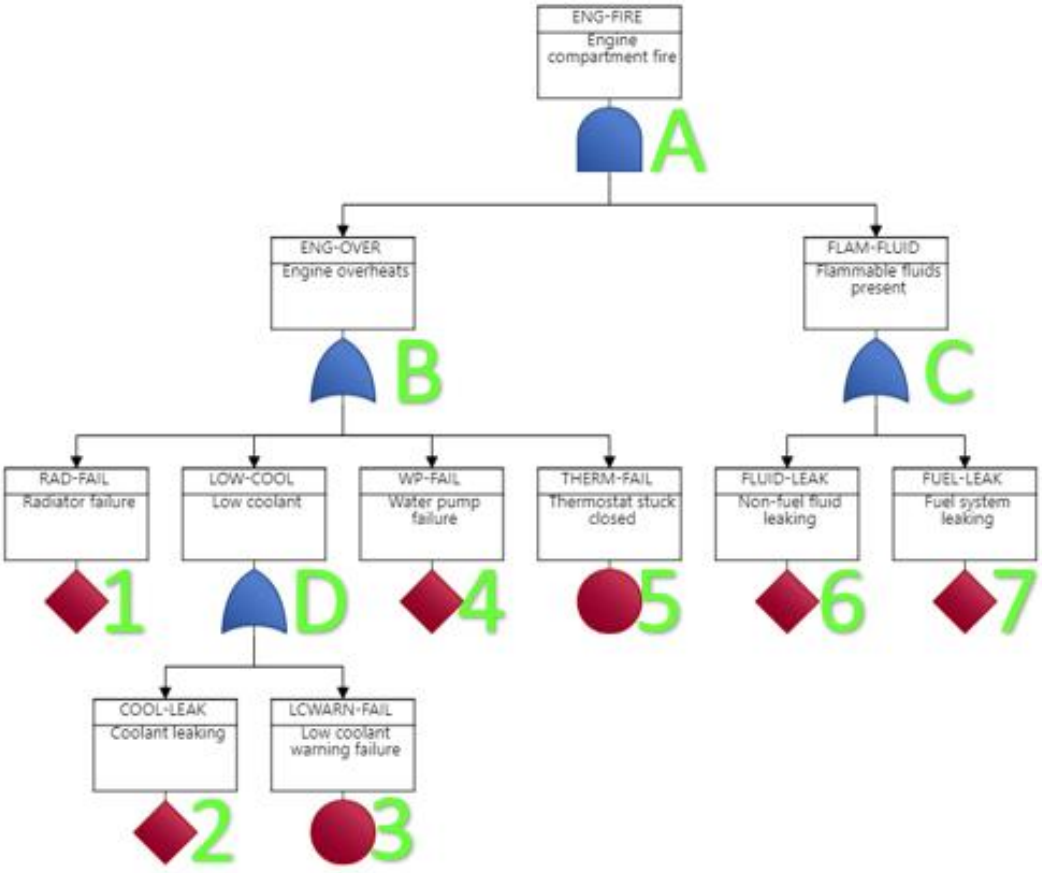
MOCUS stands for “method of obtaining cut sets” and applies a top-down approach to cut set analysis. It was proposed by Fussell and Vesely in the early 1970s and applies two key observations:

1. OR gates increase the number of cut sets
2. AND gates enlarge the size of a cut set

MICSUP stands for "minimal cut sets upwards" and applies a bottom-up approach. It was proposed by Pande, Spector, and Chatterjee in the mid-1970s.

Example MICSUP Analysis

Using the MICSUP algorithm, you can identify the minimal cut sets for our vehicle engine compartment fire fault tree. It is useful to start by identifying each of the Logic symbols with a letter and all the terminal events with a number, or whatever identification system is most helpful for you.



With MICSUP, we start at the bottom, looking at D first, which is an OR gate and so we get two cut sets, 2 alone and 3 alone.

We look to the next higher level at C, which is also an OR gate and gives us 2 cut sets – 6 alone and 7 alone.

Next, we look at B, also an OR gate and its cut sets are 1, D (which can be replaced with 2 and 3), 4 and then 5.

Finally, we look at A, which is an AND gate with a single cut set B-C, with B getting replaced by its cut sets (all single items) and C getting replaced by its cut sets, 6 or 7.

There are ten cut sets for the undesired top event, *Engine compartment fire*. They include:

- 1 – 6
- 1 – 7
- 2 – 6
- 2 – 7
- 3 – 6
- 3 – 7
- 4 – 6
- 4 – 7
- 5 – 6
- 5 – 7

All cut sets identified are minimal cut sets, so no further reduction is needed.

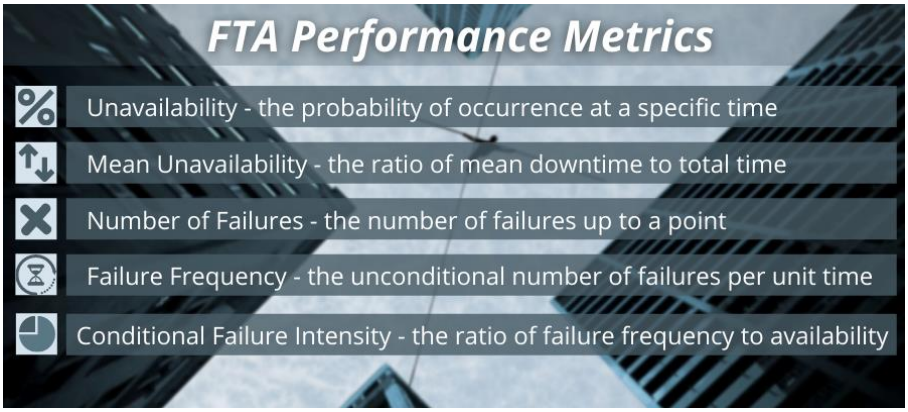
Applying a cut set generation algorithm to identify minimal cut sets is the best approach to avoid potential errors. The most efficient way to find minimal cut sets is to apply cut set generation algorithms using established fault tree analysis software, such as Relyence Fault Tree.

Calculate Metrics

Oftentimes, fault tree analysis is used to evaluate key system performance metrics, such as:

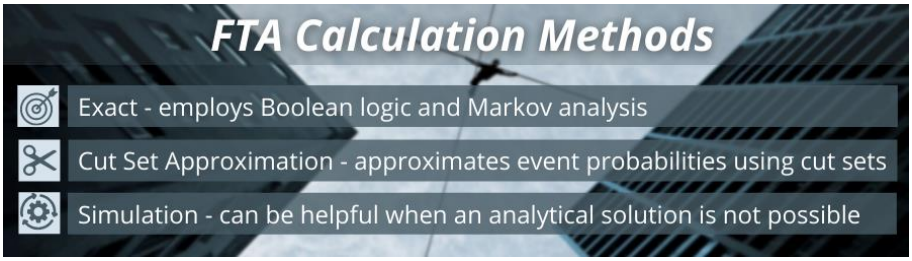
- Unavailability: The unavailability at a certain time point. The probability of occurrence at a specific time point.
- Mean Unavailability: The average unavailability. The ratio of mean downtime to total time.
- Number of failures: The expected number of failures up to a point in time.
- Failure frequency: The unconditional expected number of failures per unit time.

- Conditional failure intensity: The expected number of failures per unit time given the system is operational at the beginning of the time interval. The ratio of failure frequency to availability.



These quantitative metrics can be calculated in various ways.

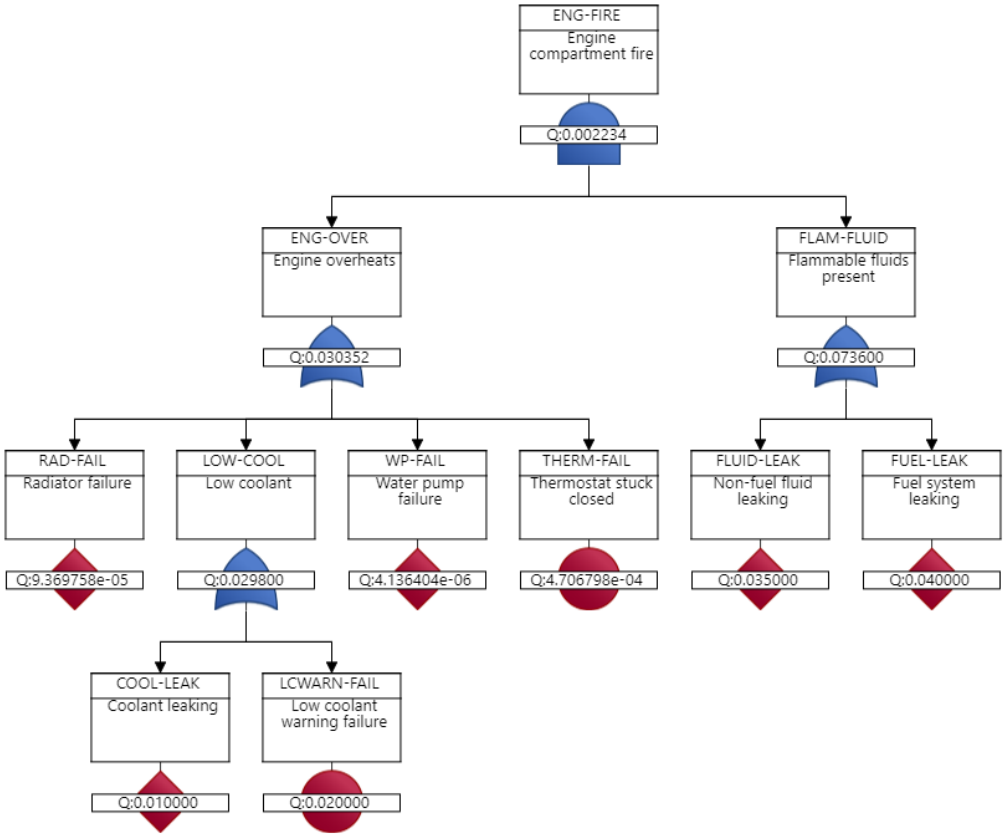
- *Exact* methods employ Boolean logic for static gates and can utilize Markov analysis for dynamic gates.
- *Cut set approximation* methods approximate the event probabilities using cut sets. As a result, cut set approximation methods work best when the terminal events have small failure probabilities. Several cut set approximation methods used for quantitative fault tree analysis include Cut Set Summation, Cross Product, and Esary Proschan.
- *Simulation*, or Monte Carlo simulation, is particularly helpful for very complex systems when an analytical solution is not possible. Simulation can also be useful to help prove analytical results.



To calculate metrics like unavailability, we need to incorporate quantitative input data for all terminal events in the fault tree. That quantitative input data can be probability values or other data such as failure rate for the terminal events to determine metrics like unavailability for the intermediate and top event.

Example Quantitative Metrics Calculation

Consider the example fault tree for Engine compartment fire. By providing quantitative input data such as constant probability or failure rate and time, we can calculate quantitative results for all terminal events, intermediate events, and the top event.



Sources for Data Needed for Quantitative Analysis

Such critical quantitative data can come from various sources. The failure rates for the various hardware failures may be available from a Reliability Prediction analysis or accepted data sources like NPRD (Non-Electronic Parts Reliability Database) and EPRD (Electronic Parts Reliability Database). Failure rates could also be defined based on the mode failure rates from FMEA. Data such as failure rates can also be sourced from test or field data analyzed with Weibull, ALT (Accelerated Life Testing) or FRACAS (Failure Reporting, Analysis, and Corrective Action System) analyses.

Analyze Importance Measures

Importance measures, or reliability importance measures, are results that can be calculated as part of fault tree analysis to identify which events, if improved (i.e., reduce their likelihood of occurrence), could give you the best improvement for system performance.

Importance measures therefore take into account the individual event probabilities as well as the top event probability.

Among the various reliability importance measures commonly considered are:

- Marginal
- Criticality
- Diagnostic
- Risk Achievement Worth
- Risk Reduction Worth

Marginal (Birnbaum) Importance Measure

The Marginal measure, also called Birnbaum, measures the increase in the probability (P) of the top event (E) due to an event (A). It is reported as the difference in the probability of E given that A did occur (probability of event A is set to 1) and the probability of E given that A did not occur (probability of event A is set to 0).

Marginal Importance Measure = P(E|P(A)=1) - P(E|P(A)=0)

It allows you to see the increase in the probability of E given the occurrence of A.

One weakness of the Marginal importance measure is that it does not directly consider the probability of event A occurring, which means you can be led to assign high importance values to events that are very unlikely to occur and thus may be difficult to improve.

Criticality Importance Measure

The Criticality measure is a modification of the Marginal importance measure that also takes into account the probability of event A. It takes the Marginal importance measure and multiplies it by the probability of A divided by the probability of E.

*Criticality Importance Measure = Marginal Importance Measure * P(A) / P(E)*

Because it also takes into account the end event occurrence, it is used to highlight events that lead to the top event occurring and are also more likely to occur and thus can reasonably be improved.

Diagnostic Importance Measure

The Diagnostic measure is the fraction of the top event (E) probability (P) that includes the event (A) occurring; or it is the probability that if the top gate occurred, the event occurred.

*Diagnostic Importance Measure = P(A) * P(E|P(A)=1) / P(E)*

Risk Achievement Worth (RAW) Importance Measure

The Risk Achievement Worth (RAW), or Top Increase Sensitivity, measure is the increase in probability of top event E when event A is given to occur. It reports the ratio of the probability of E when event A is given to occur (probability of event A is set to 1) and the probability of E.

RAW Importance Measure = P(E|P(A)=1) / P(E)

Events with the largest RAW measure values have the largest impact on the probability of the top gate, P(E), therefore it shows where prevention areas should be focused to prevent top event failures.

Risk Reduction Worth (RRW) Importance Measure

The Risk Reduction Worth (RRW), or Top Decrease Sensitivity, measure is the reduction in probability of top event E when event A is given to not occur. It reports the ratio of the probability of E and the probability of E when event A is given to not occur (probability of event A is set to 0).

RRW Importance Measure = P(E) / P(E|P(A)=0)

The RRW measure determines the maximum reduction in the top event probability if the event is improved.

More Importance Measures Analysis

For further details on importance measures, read our blog post on this topic.

Example Importance Measures Analysis

Assume you prefer to use the Diagnostic Importance Measure to help identify the event that yields the most improvement for the likelihood of occurrence of the top event. Considering the *Engine compartment fire* tree, we can calculate the Diagnostic Importance Measure for each event.

From earlier quantitative calculations, we know the P(E), where E represents the *Engine compartment fire* event, is 0.002234. For each of the contributing events, we need to identify P(A) and P(E | P(A)=1).

Event (A)	P(A)	P(E | P(A)=1)	P(E)	Diagnostic Importance Measure
RAD-FAIL	0.000094	0.073600	0.002234	0.003097
WP-FAIL	0.000004	0.073600	0.002234	0.000132
THERM-FAIL	0.000047	0.073600	0.002234	0.001548
FLUID-LEAK	0.003500	0.030352	0.002234	0.047552
FUEL-LEAK	0.040000	0.030352	0.002234	0.543456
COOL-LEAK	0.010000	0.073600	0.002234	0.329454
LCWARN-FAIL	0.020000	0.073600	0.002234	0.658908

If we sort the results from highest to lowest, we see that our best improvement for the *Engine compartment fire* event will be reduction in the likelihood of occurrence of the *Low coolant warning failure* event.

And note that utilizing a fault tree analysis tool such as Relyence Fault Tree can help you to easily calculate one or more of the reliability importance measures of interest.

Step 3: Take Steps to Improve the Safety of your Product or Process

Assuming the top event of our fault tree is an event we want to prevent given its risk to safety, the answers to how FTA can help us to develop safer systems and processes are inherently part of effective FTAs. And thus, cut set analysis, quantitative metrics and reliability importance measures can all help develop safer systems and processes.

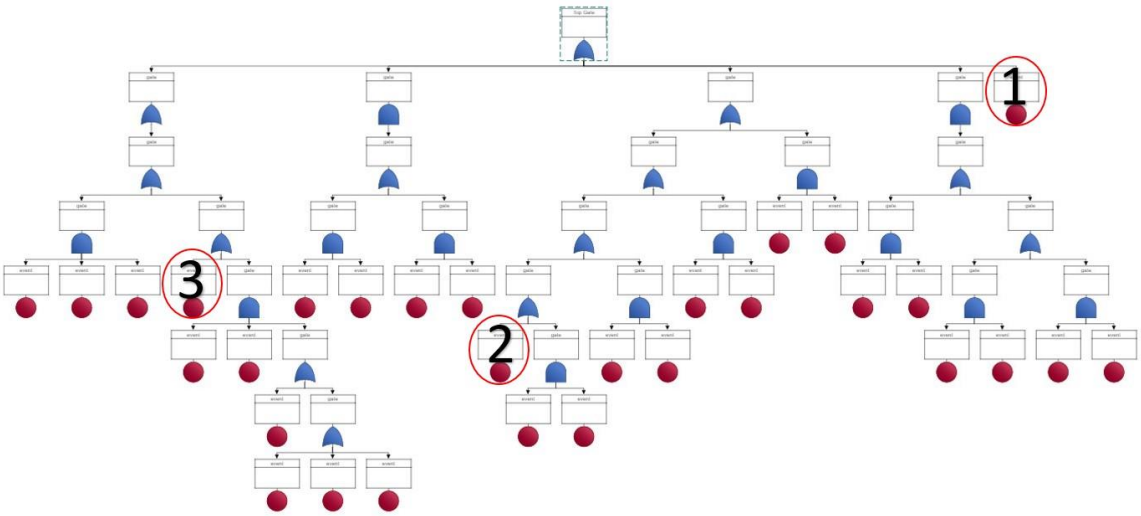
Using Minimal Cut Set Analysis to Improve Safety

Remember that minimal cut sets are the smallest group of events, such that if they all occur, cause the undesired top event to occur. There are specific circumstances where the determination of minimal cut sets can aid in safety analysis.

Identifying Single Point Failures (SPF)

In some instances, a cut set can be comprised of a single event, or a single point failure. In smaller, simpler systems you might be able to easily identify single point failures, even without minimal cut set analysis. In larger, more complex systems, single point failures could be overlooked without minimal cut set analysis.

Consider the sample fault tree pictured below. By some standards it may not seem large, but it certainly could prove useful in learning how to prevent the top undesired event.



In this sample fault tree, there are 3 example SPFs to look at as examples:

1. Consider the SPF in the top-right circled and labeled with **1**. That SPF is easy to identify, even if only using Inspection to find MCS.
2. Consider another SPF in the middle circled and labeled with **2**. It is a bit harder to find than the first one, but probably more obvious than some as we'd follow through all the OR gates. Given that its 5 levels deep in the fault tree, though, it surely could be overlooked without MCS analysis.

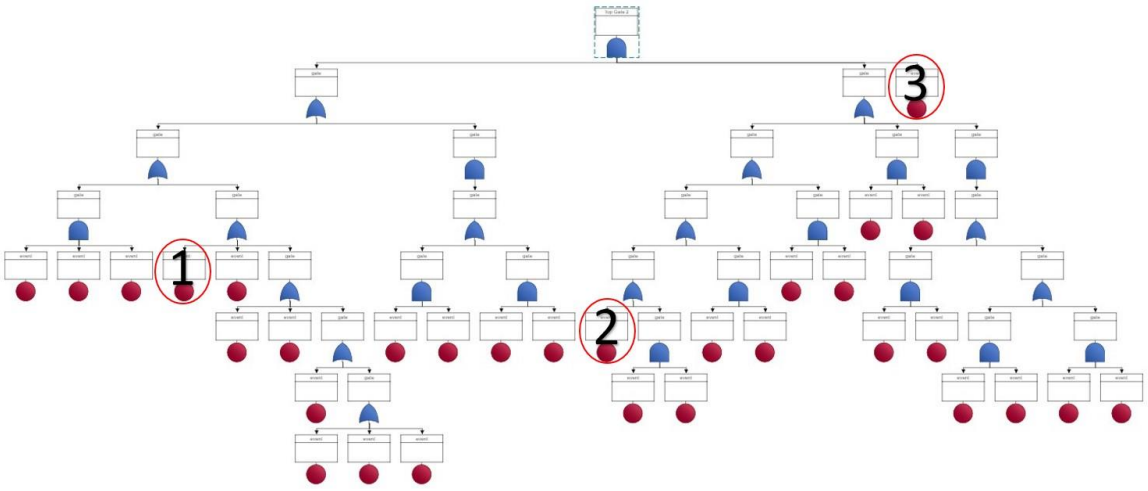
3. Consider a third SPF in the left-most branch of the fault tree circled and labeled with **3**. It might not jump out as much due to the size and structure of the fault tree, but it is indeed a SPF.

Organized, detailed cut set analysis can lead you to find these possible SPFs. You can then modify the design to avoid or reduce the likelihood of these failures to significantly improve overall safety.

Analyzing Minimal Cut Sets

Minimal cut set analysis might also help to identify cut sets that separate design teams aren't able to identify alone.

For example, consider another fault tree that models a hypothetical undesired event. Assume the left-most branch, circled and labeled with **1,** was handled by one team and the right-most branch, circled and labeled with **2**, was handled by another team. It turns out that each of those branches (which represent an intermediate event) have a single-point failure that together, along with a third single-point failure, circled and labeled with **3**, would lead to the catastrophic undesired top event.

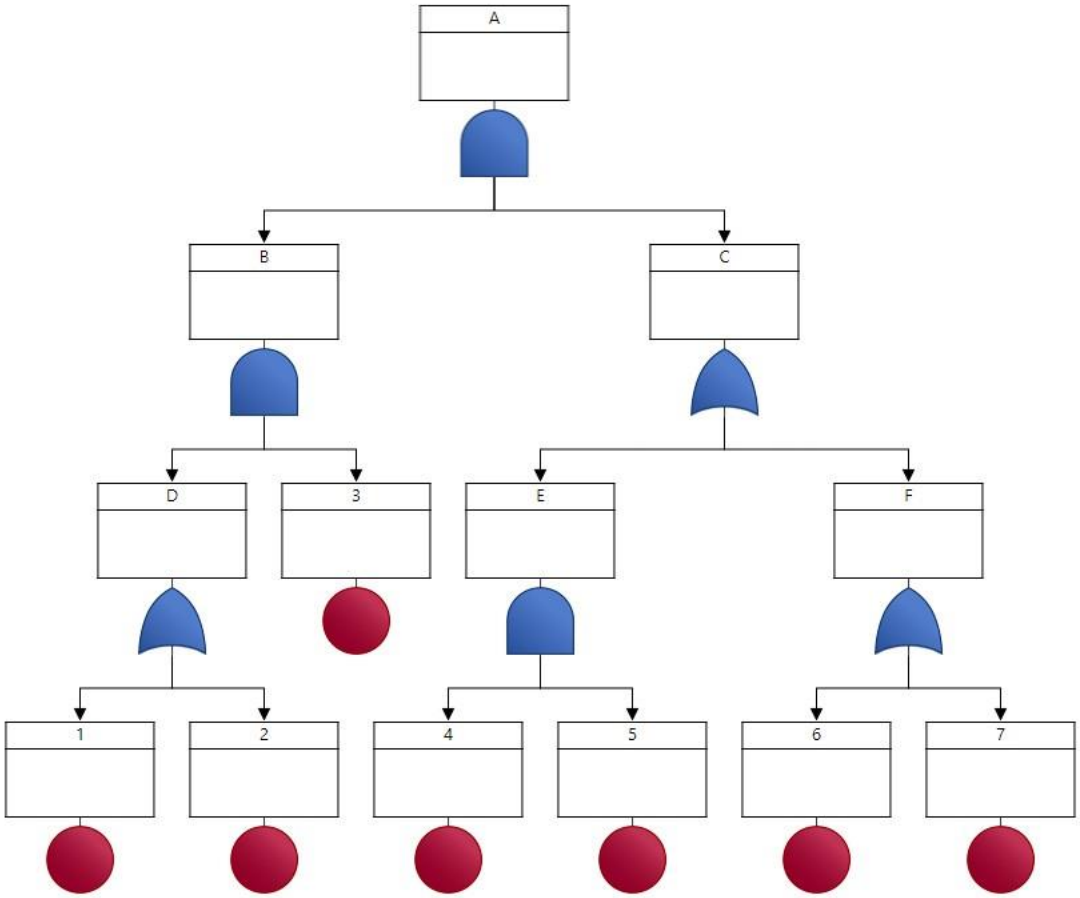


With a structured, detailed fault tree with minimal cut set analysis, this potential issue could be identified during the design process and ways to improve before it becomes a costly issue post-release could be considered.

Reviewing High Probability Cut Sets

Cut set analysis can also help you to identify the minimal cut sets with the highest probabilities.

Consider this example fault tree modeling a hypothetical undesired top event. This example fault tree has several 3rd order and 4th order cut sets.



CUT SETS

Probability	
3.600000e-009	3, 6, 2
1.800000e-009	3, 6, 1
1.200000e-010	3, 5, 4, 2
6.000000e-011	3, 5, 4, 1
4.200000e-011	3, 7, 2
2.100000e-011	3, 7, 1

We might expect the 3rd order cut sets to be more likely to occur, but depending on the probability of the input events, it might be more likely that a 4th order cut set occurs than one of the 3rd order cut sets. If this is a problem due to regulatory or design requirements, especially if it impacts safety, you can address it.

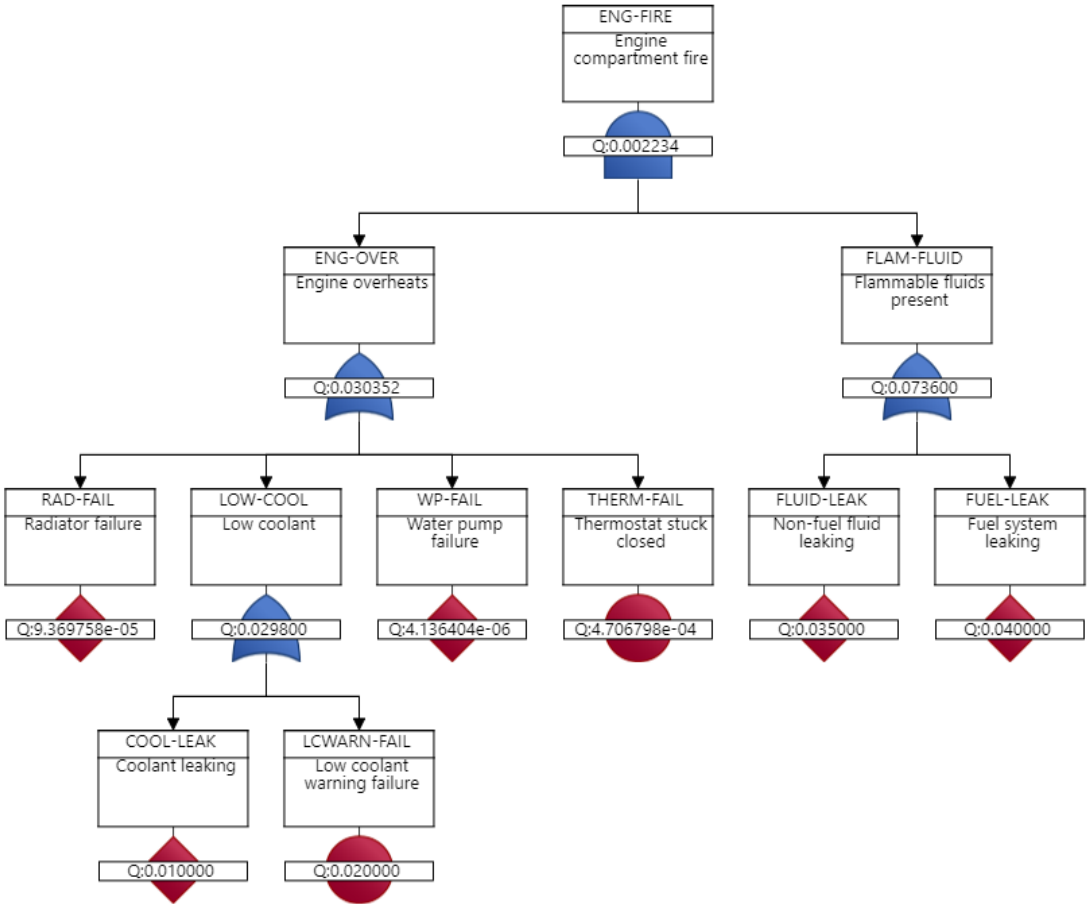
You can see that cut set analysis provides various ways for you to identify areas of concern. Armed with this knowledge, you can work to improve your system design to lower its risk profile. The value can be especially significant when performed during product design. This allows you ensure that nothing unsafe or hazardous reaches customers' hands.

Using Fault Tree Analysis Metrics to Improve Safety

As identified earlier, quantitative fault tree analysis can produce various metrics such as: unavailability (or probability of failure), mean unavailability, number of failures, failure frequency, and conditional failure intensity.

Setting goals can be important in many areas in life, especially in designing for safety. A governing body or your customer might establish an acceptable probability of risk for your design. By calculating the metrics of interest, you might find the initial design does not meet the requirement. Different configurations might be considered where redundancy is added or more highly reliability components might be incorporated. Ultimately, you can address such issues before the design is finalized.

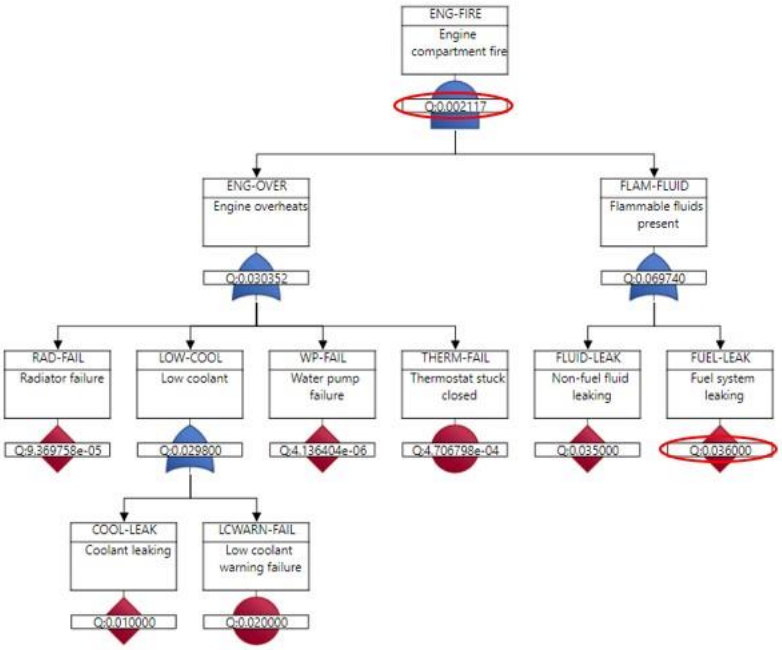
We will refer again to our example vehicle engine compartment fire fault tree to demonstrate.



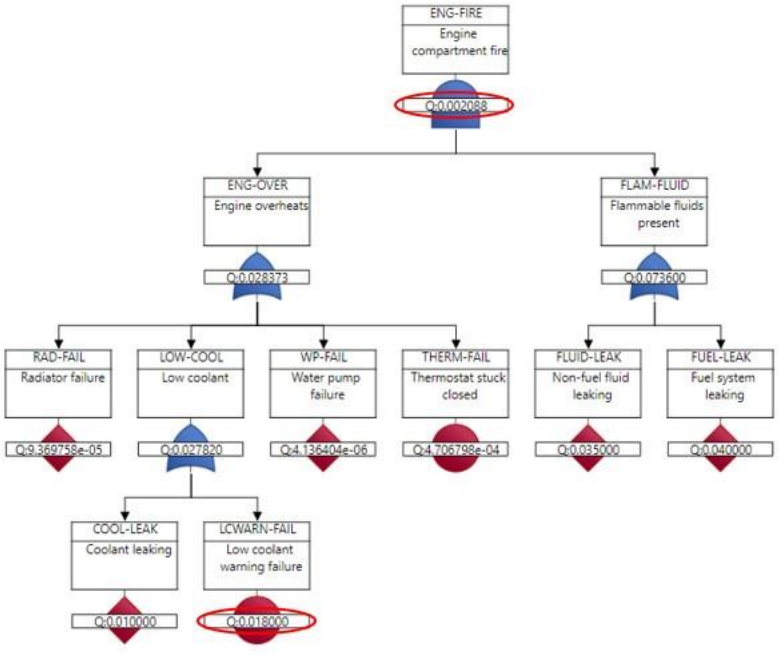
Assume the goal is to have the Unavailability < 0.002225. Based on the initial design, the Unavailability is 0.002234. We need to make some changes in order to meet our goal.

You could first look at reducing the Unavailability for any of the terminal events; essentially considering some "what if" changes:

1. Reducing the likelihood of the *Fuel system leaking* event from 0.04 to 0.036, we can reduce the unavailability of *Engine Compartment Fire* to 0.002117 – exceeding the goal.



2. Reducing the likelihood of *Low Coolant Warning Failure* from 0.02 to 0.018, the unavailability of *Engine Compartment Fire* is reduced to 0.002088, also exceeding the goal.



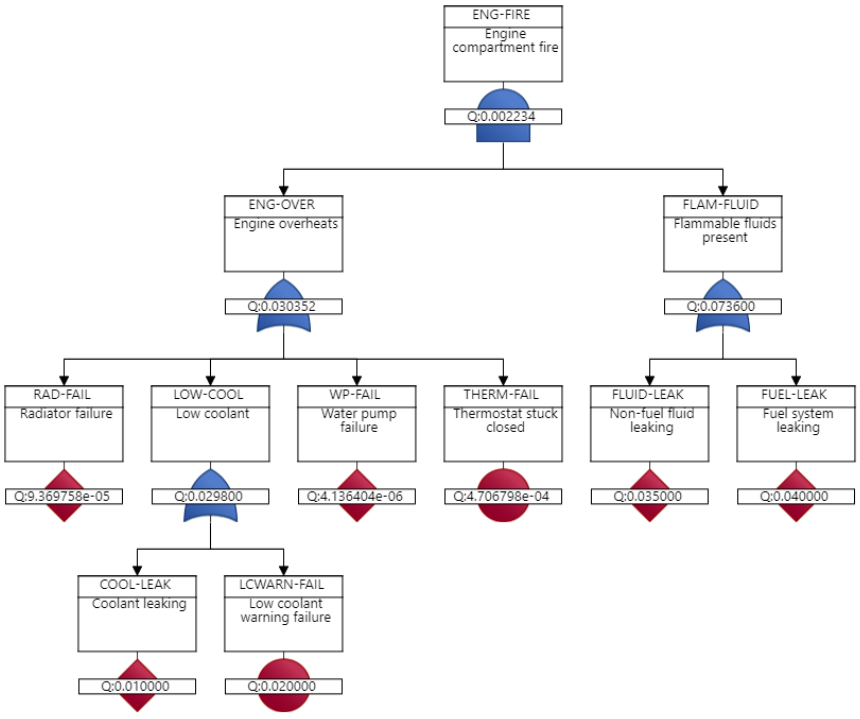
By making some simple changes to various elements of the fault tree and recalculating metrics, you can easily view the impact of various potential changes on your overall system. This enables you to quickly identify items that provide the most impact, thereby focusing your team efforts on those elements that offer the most system improvement.

Using Reliability Importance Measures to Improve Safety

Recall that reliability importance measures, are results that can be calculated as part of fault tree analysis to identify which events, if improved (i.e., reduce their likelihood of occurrence), could give you the best improvement for system performance. Individual event probabilities as well as the top event probability are considered when determining importance measures.

You can use reliability importance measures to optimize efforts and ultimately improve system safety by further reducing the likelihood of occurrence of the undesired event. Again, this lets you optimize the design, meet compliance requirements, or provide guidance on what events to focus on.

Once again, consider an example fault tree for a potential engine compartment fire.



By reviewing the results of the importance measures analysis, the optimal reduction to the likelihood of occurrence of the engine compartment fire will be by reducing the likelihood of occurrence of the *Low coolant warning failure*.

This is because if we review the importance measures calculations in the table below, where the results are sorted in descending order, we see that the calculated importance measures result is highest, or at a minimum tied for highest, for the LC-WARN FAIL (*Low coolant warning failure*) event.

IMPORTANCE MEASURES

Event	Marginal	Criticality	Diagnostic	Risk Achievement Worth	Risk Reduction Worth
LCWARN-FAIL	0.072823	0.651985	0.658945	32.947272	2.873440
FUEL-LEAK	0.029289	0.524457	0.543478	13.586957	2.102857
FLUID-LEAK	0.029137	0.456522	0.475543	13.586957	1.840000
COOL-LEAK	0.072087	0.322700	0.329473	32.947272	1.476450
THERM-FAIL	0.071400	0.015044	0.015508	32.947272	1.015274
RAD-FAIL	0.071373	0.002994	0.003087	32.947272	1.003003
WP-FAIL	0.071366	1.321474e-004	1.362832e-004	32.947272	1.000132

And, for example, if it is reduced from 0.02 to 0.018, the top event likelihood of occurrence drops to 0.002088.

In some cases, not all importance measures will offer the same results for indicating which events to focus on. This does not in any way mean that the results should be discounted. In this instance, analysts use different methods to assess the importance measures results. For example, you could take an average of all importance measures results for each event and use the highest average. Or you could identify the importance measure which most suitably applies for your situation and consider the event ranking per that specific importance measure.

Conclusion

Safety analysis is a key consideration across so many industries that design and manufacture products – including everything from highly-complex military equipment, power plants, vital medical devices, high-tech consumer products, automobiles, appliances, and so on. There are very few areas where safety is not a critical concern. Fault Tree Analysis is a simple, yet powerful technique to help design and manufacture more reliable and safer systems and processes.

Organizations have much to gain by using FTA for safety analysis and other analysis purposes. Fault Tree Analysis can also be used to improve overall reliability and availability, to gain a better understanding of your system or process, to debug complex systems in software development, and to evaluate system performance in a variety of ways.

Relyence Fault Tree is a powerful and easy-to-use tool for performing all the analysis techniques described in this paper. It also offers an array of additional features for efficient FTA. For more information, contact us or try it out for free.