

# Fault Tree Analysis

**LEVEL – 1 TRAINING**

**Two Days**

QUALITY

© 2021, Omnex, Inc.  
315 Eisenhower Parkway Suite 214  
Ann Arbor, Michigan 48108  
USA  
734-761-4940  
Fax: 734-761-4966

**Third Edition**  
**SEP 2021**

*This publication is protected by Federal Copyright Law, with all rights reserved. No part of this publication may be stored in a retrieval system, transmitted or reproduced in any way, including but not limited to photocopy, photograph, magnetic or other record, without the prior agreement and written permission of the publisher.*

QUALITY



Omnex provides training, consulting and software solutions to the international market with offices in the USA, Canada, Mexico, China (PRC), Germany, India, the Middle East, and SE Asia. Omnex offers over 400 standard and customized training courses in business, quality, environmental, food safety, laboratory and health & safety management systems worldwide.

**Email:** [info@omnex.com](mailto:info@omnex.com)

**Web:** [www.omnex.com](http://www.omnex.com)

QUALITY



# Course Objectives

- Understand the purpose of FTA.
- Understand the different symbol used in FTA.
- Demonstrate an ability to construction and effectively complete the FTA.
- Integration of FMEA with FTA Concept.
- Developing FMEA by using FTA.

QUALITY



# Agenda

- Chapter 1 – Introduction of FTA
- Chapter 2 – Understanding symbols of FTA
- Chapter 3 – Development of FTA
  - Breakout Exercise 1: Create a fault tree.
  - Breakout Exercise 2: Create FMEA using FTA.
  - Breakout Exercise 3: Probabilistic Risk Assessment.

QUALITY

# A BRIEF INTRODUCTION TO OMNEX



QUALITY



# Omnex Introduction

- International consulting, training and software development organization founded in 1985.
- Specialties:
  - Integrated management system solutions.
  - Elevating the performance of client organizations.
  - Consulting and training services in:
    - Quality Management Systems, e.g., ISO 9001, IATF 16949, AS9100, QOS.
    - Environmental Management Systems, e.g., ISO 14001.
    - Health and Safety Management Systems, e.g., ISO 45001.
- Leader in Lean, Six Sigma and other breakthrough systems and performance enhancement.
  - Provider of Lean Six Sigma services to Automotive Industry via AIAG alliance.



# About Omnex

- Headquartered in Ann Arbor, Michigan with offices in major global markets.
- In 1995-97 provided global roll out supplier training and development for Ford Motor Company.
- Trained more than 100,000 individuals in over 30 countries.
- Workforce of over 400 professionals, speaking over a dozen languages.
- Former Delegation Leader of the International Automotive Task Force (IATF) responsible for ISO/TS 16949.
- Served on committees that wrote QOS, ISO 9001, QS-9000, ISO/TS 16949 and its Semiconductor Supplement, and ISO IWA 1 (ISO 9000 for healthcare).
- Member of AIAG manual writing committees for FMEA, SPC, MSA, Sub-tier Supplier Development, Error Proofing, and Effective Problem Solving (EPS).



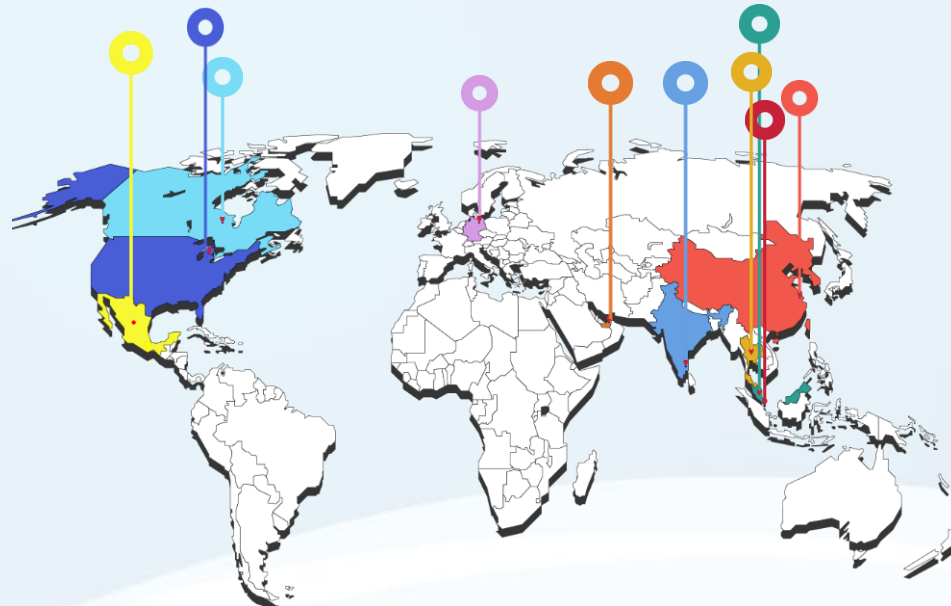














Omnex is headquartered and operates from the United States through offices in Michigan.

The company maintains international operations in many countries to provide comprehensive services to clients throughout Western Europe, Latin America and the Pacific Rim.

[www.omnex.com](http://www.omnex.com)  
[info@omnex.com](mailto:info@omnex.com)



- |  |  |
|--|--|
|  Omnex Global Head Quarters (Michigan, USA)<br>West Coast Operations (San Jose, CA) |  Middle East (Dubai, Saudi Arabia, Bahrain) |
|  Asia Pacific HQ (Chennai, Pune, Delhi, Bangalore)                                  |  Thailand (Bangkok)                         |
|  China (Shanghai, Guangzhou, Wuhan, Chengdu)                                      |  Mexico (Monterrey)                       |
|  Canada (Mississauga)   |  Singapore                                |
|  Europe (Berlin, Germany)   |  Malaysia (Kuala Lumpur)                  |



# Rules of the Classroom

- Start and end on time
- Return from breaks and lunch on time
- All questions welcome
- Your input is valuable and is encouraged . OMNEX
- Don't interrupt others
- One meeting at a time
- Listen – and respect others' ideas
- No “buts” – keep an open mind
- Cell phones & pagers off or silent mode
- No e-mails, texting or tweeting during class
- If you must take a phone call or answer a text please leave the room for as short a period as possible



# Icebreaker

- Instructor Information:
  - Name
  - Background
- Student Introductions:
  - Name
  - Position / Responsibilities
  - What is your involvement in FTA?
  - What are your experiences with respect to FTA?
  - Please share something unique and/or interesting about yourself.



QUALITY

# Chapter 1

## Introduction of Fault Tree Analysis

QUALITY

# The Thought Process

- “To design systems that work correctly we often need to understand and correct how they can go wrong.”
- Dan Goldin, NASA Administrator, 2000
- FTA identifies, models and evaluates the unique interrelationship of events leading to :
  - Failure
  - Undesired Events / States
  - Unintended Events / States

QUALITY

# Introduction of Fault Tree Analysis

- **The Beginning Years (1961 – 1970)**
  - H. Watson of Bell Labs, along with A. Mearns, developed the technique for the Air Force for evaluation of the Minuteman Launch Control System, circa 1961.
  - Recognized by Dave Haasl of Boeing as a significant system safety analysis tool (1963).
  - First major use when applied by Boeing on the entire Minuteman system for safety evaluation (1964 – 1967, 1968-1999).
  - The first technical papers on FTA were presented at the first System Safety Conference, held in Seattle, June 1965.
  - Boeing began using FTA on the design and evaluation of commercial aircraft, circa 1966.
  - Boeing developed a 12-phase fault tree simulation program, and a fault tree plotting program on a Calcomp roll plotter.
  - Adopted by the Aerospace industry (aircraft and weapons).

QUALITY





# Introduction of Fault Tree Analysis

- The Early Years (1971 – 1980)
- Adopted by the Nuclear Power industry.
- Power industry enhanced codes and algorithms
- Some of the more recognized software codes include:
  - Prepp/Kitt, SETS, FTAP, Importance and COMCAN
- The Present (1981 – 1999)
- Usage started becoming international, primarily via the Nuclear Power industry.
- More evaluation algorithms and codes were developed.
- A large number of technical papers were written on the subject (codes & algorithms).
- Usage of FTA in the software (safety) community.
- Adopted by the Chemical, Robotics and Software Industry.

# What is Fault Tree Analysis

- FTA maps the relationship between the faults, subsystems and redundant safety design elements by creating a logic diagram.
- Logic diagrams and Boolean algebra are used to identify the cause of the top event.
- Fault tree is the logical model of the relationship of the undesired event to more basic events.
- The top event of the fault tree is the undesired event.

QUALITY





# What is Fault Tree Analysis

- The middle events are the intermediate events and basic events are at the bottom.
- The logic relationship of events are shown by logic symbols or gates.
- Probability of occurrence values are assigned to the lowest events in the tree in order to obtain the probability of the occurrence of the top event.

QUALITY

# Why to perform the FTA?

- FTA depicts the risk based path to a root cause or base level event.
- The identified risk drive actions which are intended to mitigate the risk prior to program launch.
- Alternatively when investigating a failure, the chain of events depicted by FTA allows the problem solver to see the events leading to a root cause(S) or base level event.

QUALITY

# When to use FTA?

- Engineers are asked to anticipate the failures in advance of a product development.
- Potential failures must be identified early in the product development cycle to successfully mitigate the risk.
- This failure prevention activity is intended to protect the customer from an unacceptable experience.
- There are many tools used to identify potential failure and their cause.
- One of these tools is Fault Tree Analysis.

QUALITY



# When to use FTA?

- Root Cause Analysis
  - Identify all relevant events and conditions leading to Undesired Event
  - Determine parallel and sequential event combinations
  - Model diverse/complex event interrelationships involved. o.m.n.e.x
- Risk Assessment
  - Calculate the probability of an Undesired Event (level of risk)
  - Identify safety critical components/functions/phases
  - Measure effect of design changes
- Design Safety Assessment
  - Demonstrate compliance with requirements
  - Shows where safety requirements are needed
  - Identify and evaluate potential design defects/weak links
  - Determine Common Mode failures

- Reliability
- Availability
- Maintainability
- Safety

QUALITY



# FTA Application – When can be used

- Required by customer.
- Necessitated by the risk involved with the product (risk is high).
- Accident/incident/anomaly investigation.
- To make a detailed safety case for safety critical system.
- To evaluate corrective action or design options.
- Need to evaluate criticality, importance, probability and risk.
- Need to know root cause chain of events.
- To evaluate the effect of safety barriers.
- Determine best location for safety devices (weak links).

QUALITY



# FTA Strength

- Visual model -- cause/effect relationships.
- Easy to learn, do and follow.
- Models complex system relationships in an understandable manner
  - Follows paths across system boundaries
  - Combines hardware, software, environment and human interaction
- Probability model.
- Scientifically sound.
  - Boolean Algebra, Logic, Probability, Reliability
  - Physics, Chemistry and Engineering
- Commercial software is available.
- FT's can provide value despite incomplete information.
- Proven Technique.

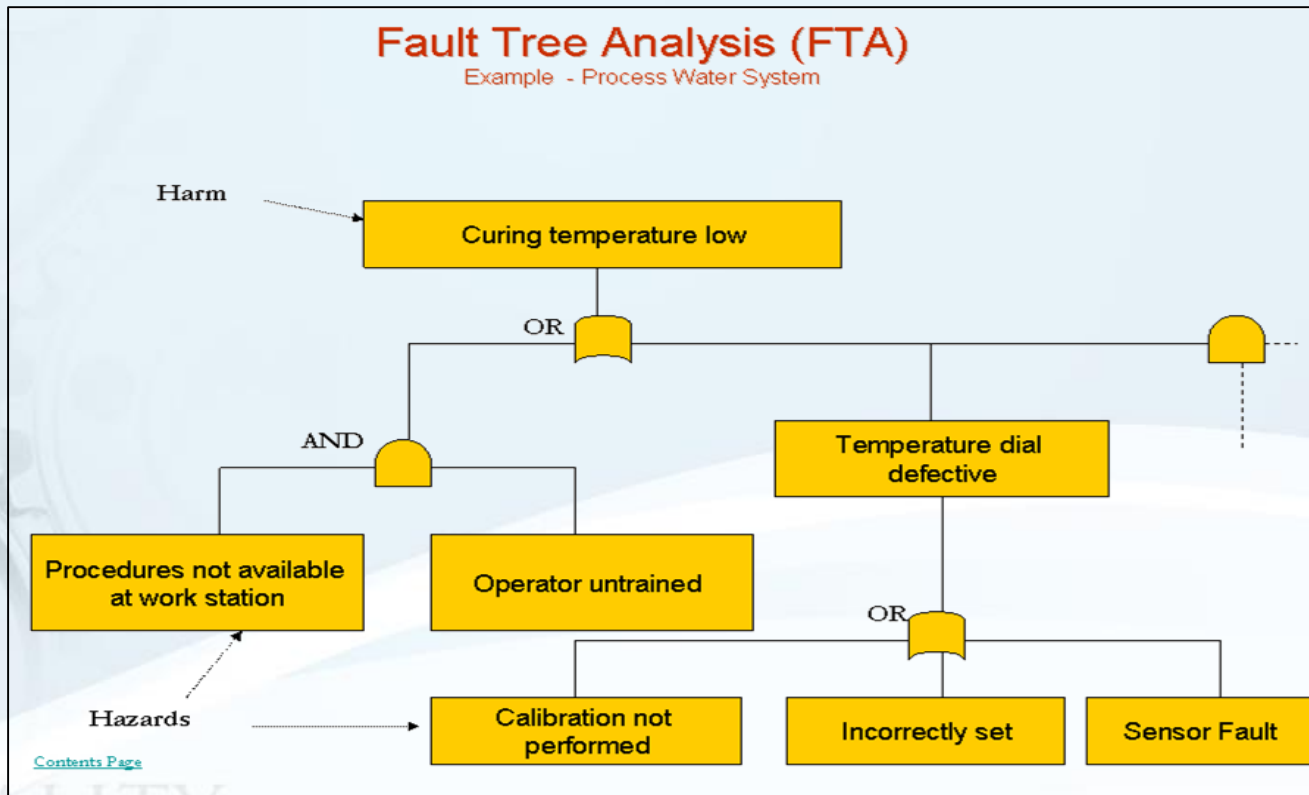
# Example for FTA Application

- Damping force low.
- AC not cooling.
- Axle welding crack.
- Unintended deployment of air bag.
- Seat belt failure.
- Failure of Electrical control Unit.
- Evaluate the accidental operation and crash of a railroad car.
- Evaluate spacecraft failure.
- Calculate the probability of a torpedo striking target vessel.
- Evaluate a chemical process and determine where to monitor the process and establish safety controls.
- Calculate the probability of a plant accident.

QUALITY

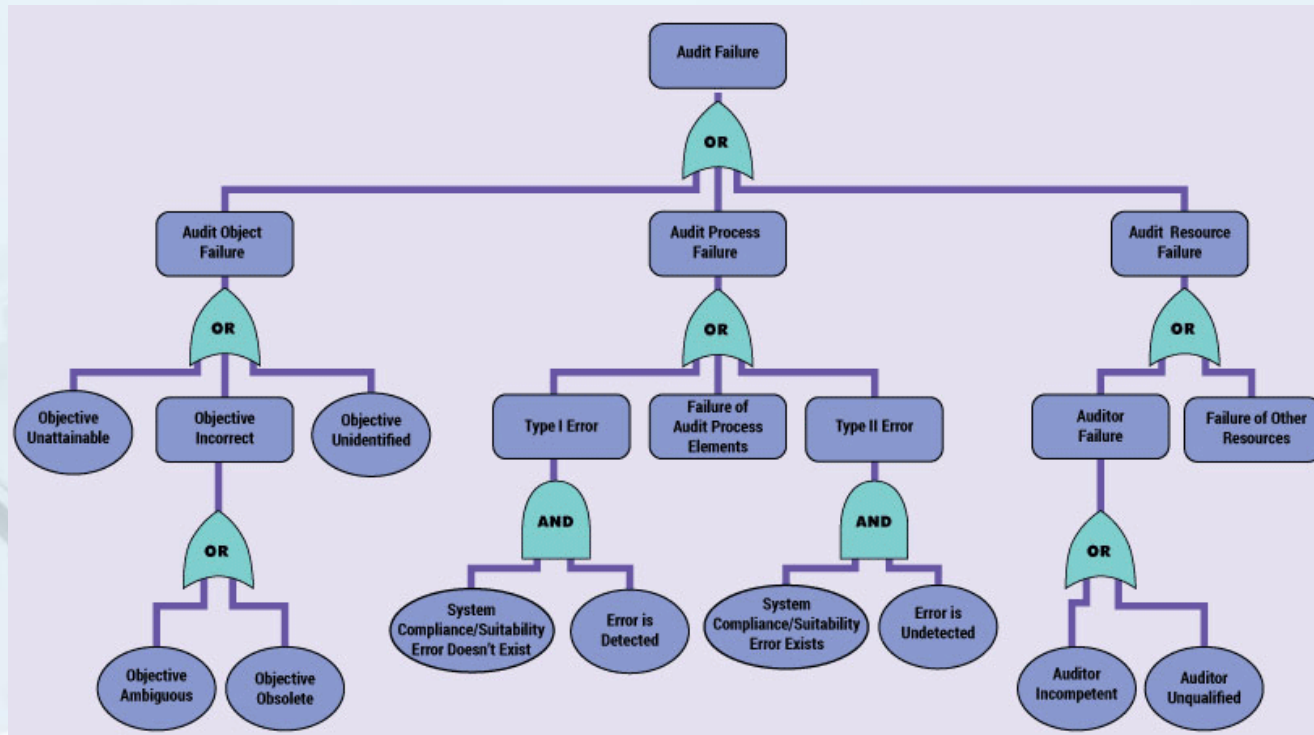


# Fault Tree Analysis – Example 1





# Fault Tree Analysis – Audit Failure Example 2



QUALITY

# Preventive Approach

- With the preventive application of the FTA the focus is on the hazard or risk analysis.
- Objectives of preventive use of the FTA:
  - Minimize design errors (malfunction, non-function),
  - Verify and demonstrate the system safety,
  - Increase the system reliability and
  - Assess potential risks within the scope of risk management.

QUALITY



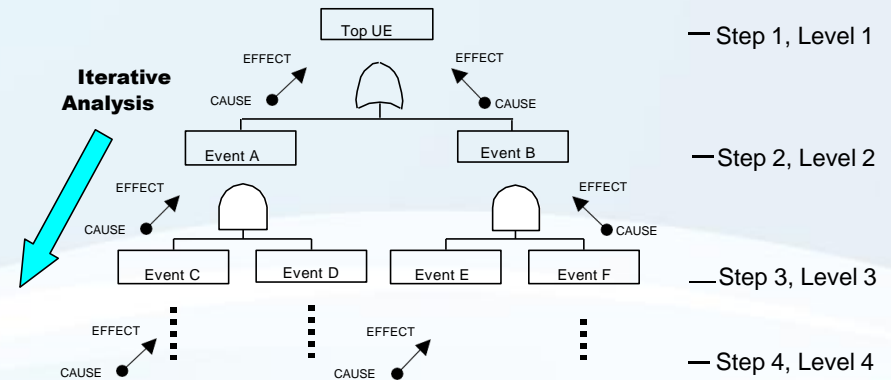
# Corrective Approach

- The corrective approach of the FTA is used in the damage analysis or risk analysis.
- In case of analyzing damages the FTA serves as decision-making assistance for legal issues(e.g. product liability) and also in determining the risks for the risk-management process.
- The results from the analysis are used to assess a damage or accident sequence.

QUALITY

# FTA Philosophy

- Review the Gate Event under investigation
- Identify all the possible causes of this event
- Ensure you do not jump ahead of a possible cause event
- Identify the relationship or logic of the Cause-Effect events
- Structure the tree with these events and logic gate
- Keep looking back to ensure identified events are not repeated
- Repeat the process for the next gate.



QUALITY

# FTA Pitfalls

- **Lack of proper FT planning and design can result in problems**
  - Might necessitate restructure of entire tree.
  - Might necessitate renaming all events in tree.
  - Rework will cost time and money.
- **Must plan ahead**
  - Leave room for future tree expansion
  - Allow for possible future changes in tree without repercussions
  - Structure tree carefully, later changes can impact entire tree
- **Large FT's require more design foresight**
  - Develop organized plan when several analysts work on same FT

QUALITY



# Roles and Responsibilities

- **FTA coordinator:** The FTA coordinator is the representative of the method for a unit .
- **FTA expert:** The FTA expert is responsible for the moderation on the FTA team.
- **FTA team:** The FTA team is made up of the FTA experts and the relevant specialists.
- **FTA contractor:** The FTA contractor defines the goals and the scope of the considerations for the FTA to be conducted. The contractor is the “Sponsor” of the FTA and ensures the budget and the resources for the work
- **FTA reviewer:** The FTA reviewer is responsible for reviewing the contents of a FTA before it is released.

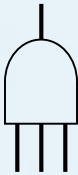

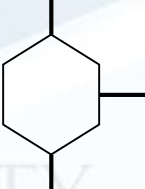
# Chapter 3

## Understand the symbol of FTA

QUALITY

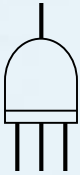

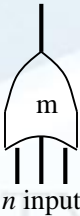


# Basic Fault Tree Structure – Gate Symbols

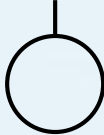
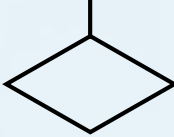
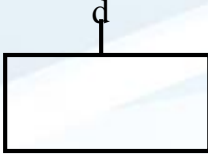
	Gate Symbol	Gate Name	Causal Relation
1		<b>AND</b> gate	Output event occurs if all input events occur simultaneously.
2		<b>OR</b> gate	Output event occurs if any one of the input events occurs.
3		<b>Inhibit</b> gate	Input produces output when conditional event occurs.



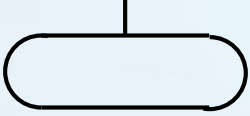
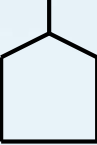

# Basic Fault Tree Structure – Gate Symbols

	Gate Symbol	Gate Name	Causal Relation
4		<b>Priority AND gate</b>	Output event occurs if all input events occur in the order from left to right.
5		<b>Exclusive OR gate</b>	Output event occurs if one but not both, of the input events occurs.
6		<b><i>m</i> Out of <i>n</i> gate (voting or sample gate)</b>	Output event occurs if <i>m</i> out of <i>n</i> input events occur.

# Basic Fault Tree Structure – Event Symbols

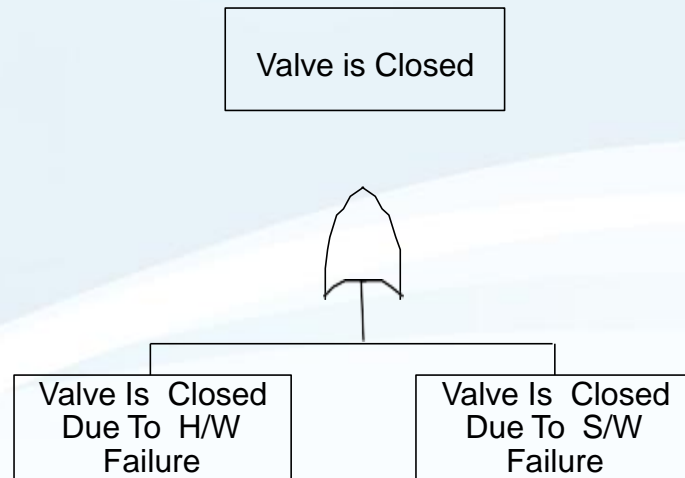
	Event Symbol	Meaning of Symbols
1	 Circle	Basic event with sufficient data <b>Root cause (= basic fault)</b> <b>(e.g. part failure, software error, human error)</b> <b>Basic Event – A lower most event that can not be further developed.</b>
2	 Diamon	Undeveloped event <b>An event (Fault) which has scope for further analyzed/developed with more time or information but not done usually because of insufficient data.</b>
3	 Rectangl	Event represented by a gate <b>They are a logical combination of lower level event.</b>

# Basic Fault Tree Structure – Event Symbols

	Event Symbol	Meaning of Symbols
4	 Oval	Conditional event used with inhibit gate
5	 House	House event. Either occurring or not occurring
6	 Triangles	Transfer symbol

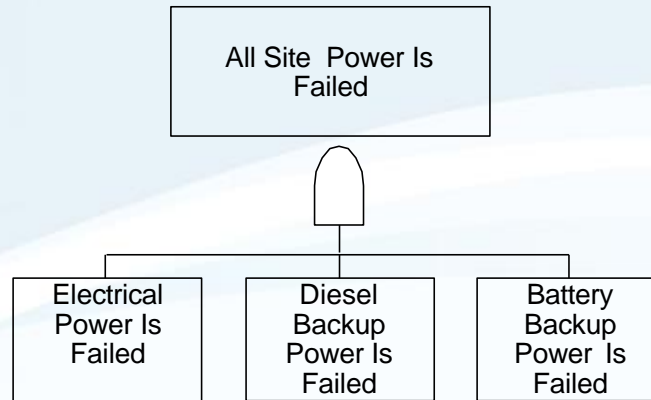
# OR Gate

- Causality never passes through an OR gate
  - The input faults are never the cause of the output fault
  - Inputs are identical to the output, only more specifically defined (refined) as to cause



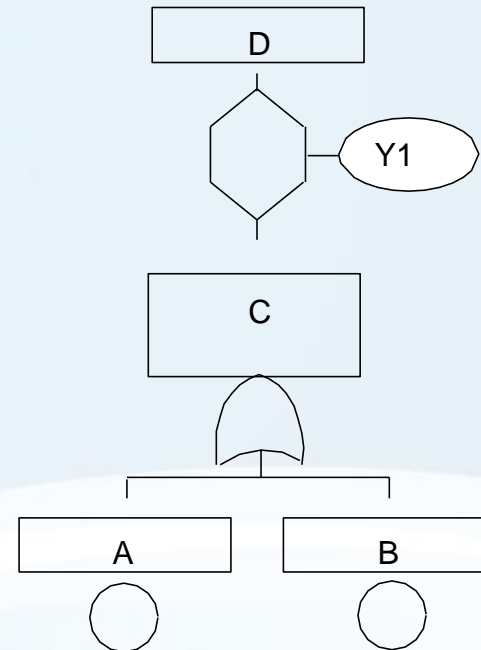
# AND Gate

- Specifies a causal relationship between the inputs and the output
  - The input faults collectively represent the cause of the output fault.
  - Implies nothing about the antecedents of the input faults.



# Inhibit Gate

- Both C and Y1 are necessary to cause D
- Y1 is a condition or probability
- Pass through if condition is satisfied
- Essentially an AND gate



QUALITY

# Chapter 3

## Development of FTA

QUALITY

# Steps to conduct FTA?

Step 1

- Define the undesired event to study

Step 2

- Obtain an understanding of the system

Step 3

- Construct the fault tree

Step 4

- Evaluate the fault tree

Step 5

- Control the hazards(undesired Event) identified



# STEP 1

Define the undesired event to study

QUALITY O-M-N-E-X

# Identify the Undesired Event or Hazard

Knowing the consequence of the failure is useful in defining the Top-level event of the Fault Tree. The Top-level event, or Hazard, should be defined as precisely as possible:

How much?

How long (duration)?

What is the safety impact?

What is the environmental impact?

What is the regulatory impact?

QUALITY

# Define the top Undesired Event

- Purpose
  - The analysis starts here, shapes entire analysis
  - Very important, must be done correctly
- Start with basic concern
  - Hazard, requirement, safety problem, accident/incident
- Define the UE in a long narrative format
- Describe UE in short sentence
- Test the defined UE
- Determine if UE is achievable and correct
- Obtain concurrence on defined UE

QUALITY

# Example of To UE's

- Inadvertent Weapon Unlock
- Inadvertent Weapon Release
- Incorrect Weapon Status Signals
- Failure of the MPRT Vehicle Collision Avoidance System
- Loss of All Aircraft Communication Systems
- Inadvertent Deployment of Aircraft Engine Thrust Reverser
- Offshore Oil Platform Overturns During Towing
- Loss of Auto Steer-by-wire Function

QUALITY

# STEP 2

Obtain an understanding of the system

QUALITY

# Define the system

- Obtain system design information
  - Drawings, schematics, procedures, timelines
  - Failure data, exposure times
  - Logic diagrams, block diagrams, IELs
- Know and understand
  - System operation
  - System components and interfaces
  - Software design and operation
  - Hardware/software interaction
  - Maintenance operation
  - Test procedures

**Guideline -- If you are unable to build block diagram of the system, your understanding may be limited.**

QUALITY

# Obtain understanding of system being analysed?

Create or acquire appropriate support information:

- List of components (Bill of Material)
- Boundary Diagram
- Schematic
- Code Requirements
- Engineering Noises and Environments
- Examples of similar products or failures

QUALITY



# Obtain understanding of system being analysed?

- List the potential causes of the hazard to the next level. This is similar to the **Why-Why analysis** process, except development of a Fault Tree should be focused on a single level before progressing to the next.
- Include system design engineers, who have full knowledge of the system and its functions, in the higher levels of the Fault Tree Analysis. This knowledge is very important for cause selection.

QUALITY

# Obtain understanding of system being analysed?

- Include Reliability Engineers who can assist in developing the relationships of causes to a failure or fault.
- Estimate probability of the causes at the Base-level event
- Label all causes with codes (optional)
- Prioritize or sequence causes in the order of occurrence or probability

QUALITY

# Establish the Boundary

- Define the analysis ground rules
- Define assumptions
- Bound the overall problem
- Obtain concurrence
- Document the ground rules, assumptions and boundaries

## Boundary Factors

- System performance – areas of impact
- Size – depth and detail of analysis
- Scope of analysis – what subsystems and components to include
- System modes of operation – startup, shutdown, steady state
- System phase(s)
- Available resources (i.e., time, dollars, people)
- Resolution limit (how deep to dig)
- Establish level of analysis detail and comprehensiveness

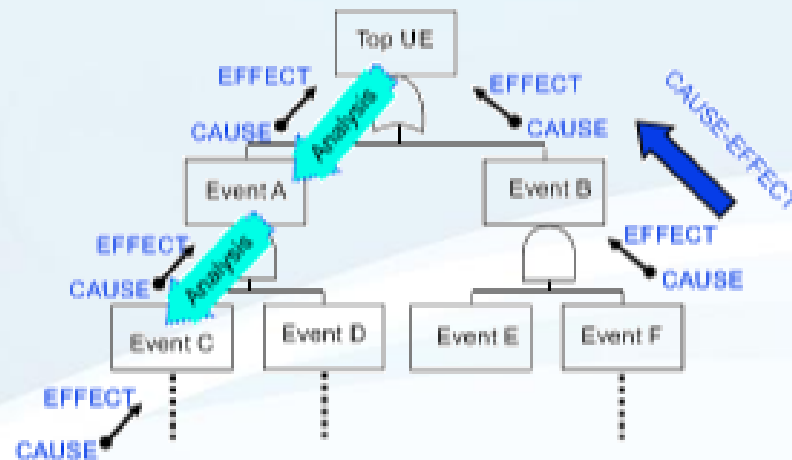
# STEP 3

Construct a Fault Tree

QUALITY

# Develop the FTA

- Follow rules and definitions of FTA
- Iterative process
- Continually check against system design
- Continually check ground rules
- Tree is developed in layers, levels and branches

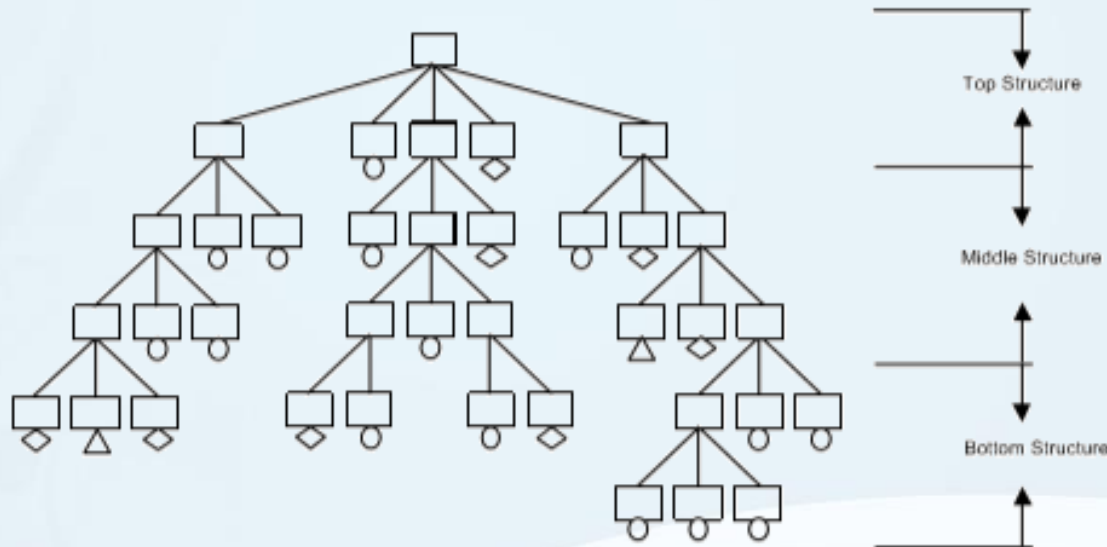


QUALITY

# FTA Construction process

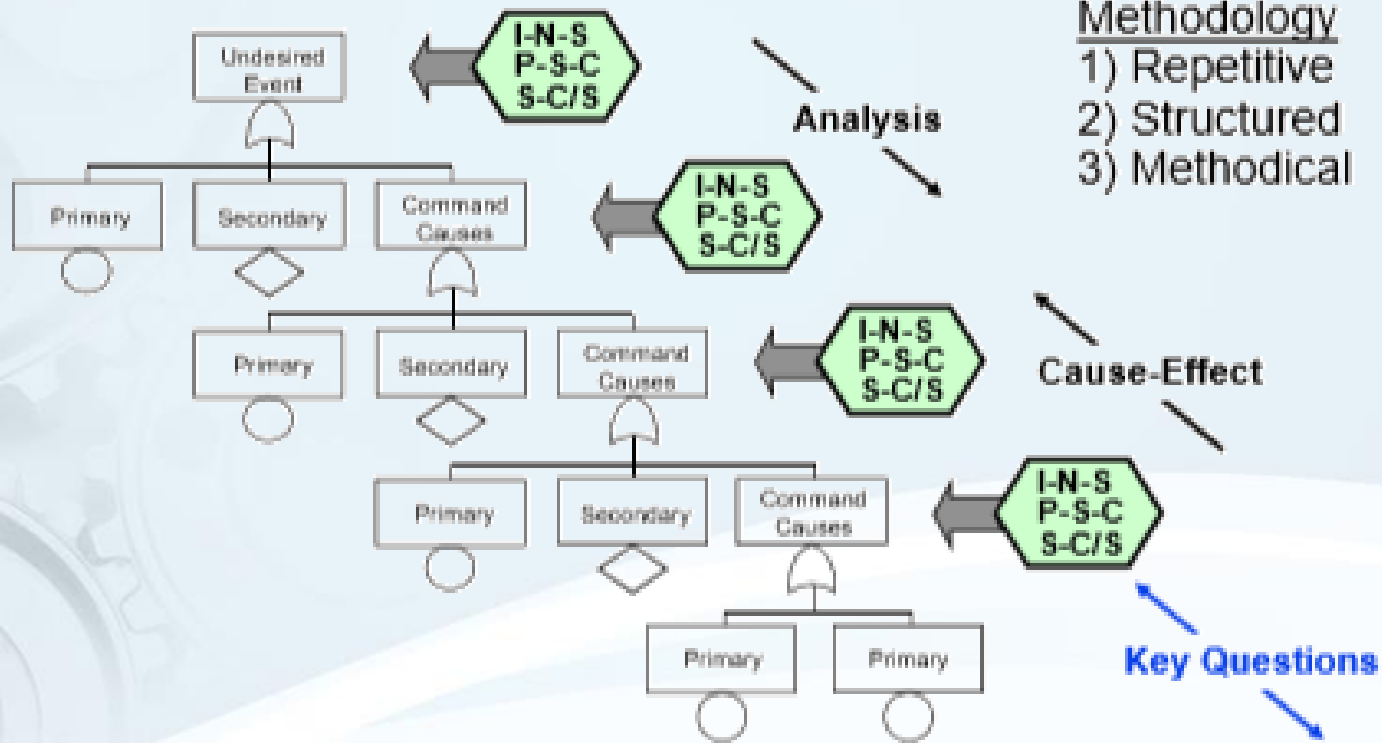
- Tree is developed in:
  - Layers
  - Levels
  - Branches
- Tree Levels:
  - **Top** Level
    - ◆ Defines the top in terms of discrete system functions that can cause the top UE
    - ◆ Shapes the overall structure of the tree
  - **Intermediate** Level
    - ◆ Defines the logical relationships between system functions and component behavior
    - ◆ Function – systems – subsystems – modules - components
  - **Bottom** Level
    - ◆ Consists of the Basic Events or component failure modes

# Construction Process - Overview



- Tree is developed in Layers, Levels, and Branches
- Levels represent various stages of detail
  - Top - shapes tree, combines systems
  - Middle - subsystems, functions, phases, fault states
  - Bottom - basic events, component failures

# FTA construction



I-N-S=Immediate, Necessary, Sufficient  
 P-S-C=Primary, Secondary, Command  
 S-C/S=State of the Component or System

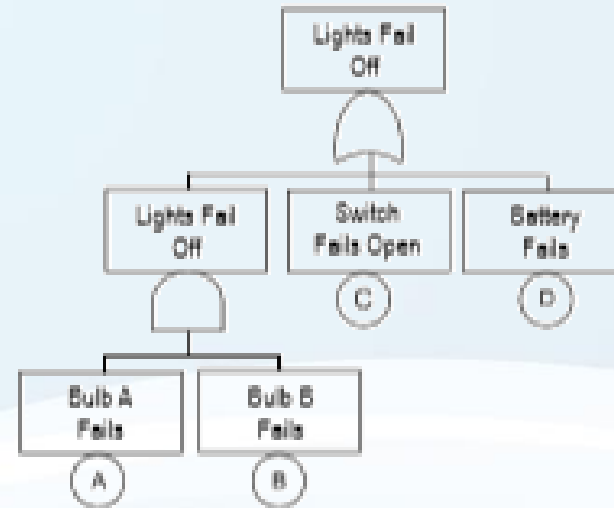
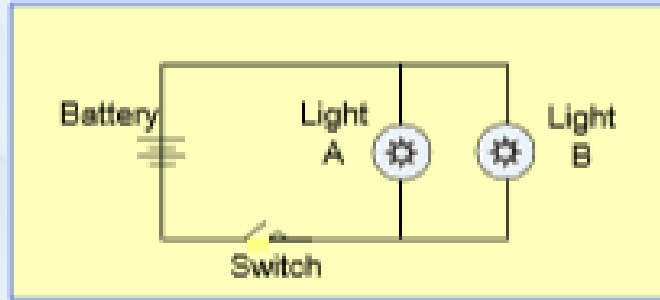


# Four basic approach of FTA

- **Component**
  - Immediately focuses on components
  - "Shopping list" approach
  - Can overlook detailed causes
- **Subsystem**
  - Immediately emphasizes subsystems
  - Can overlook detailed causes
  - Can use Functional flow method after subsystem breakdown
- **Scenario**
  - Breaks down UE into fault scenarios before detailed design analysis
  - Sometimes necessary at FT top level for complex systems
- **Functional Flow**
  - Follows system functions (command path)
  - More structured
  - Less likely to miss detail causes

QUALITY

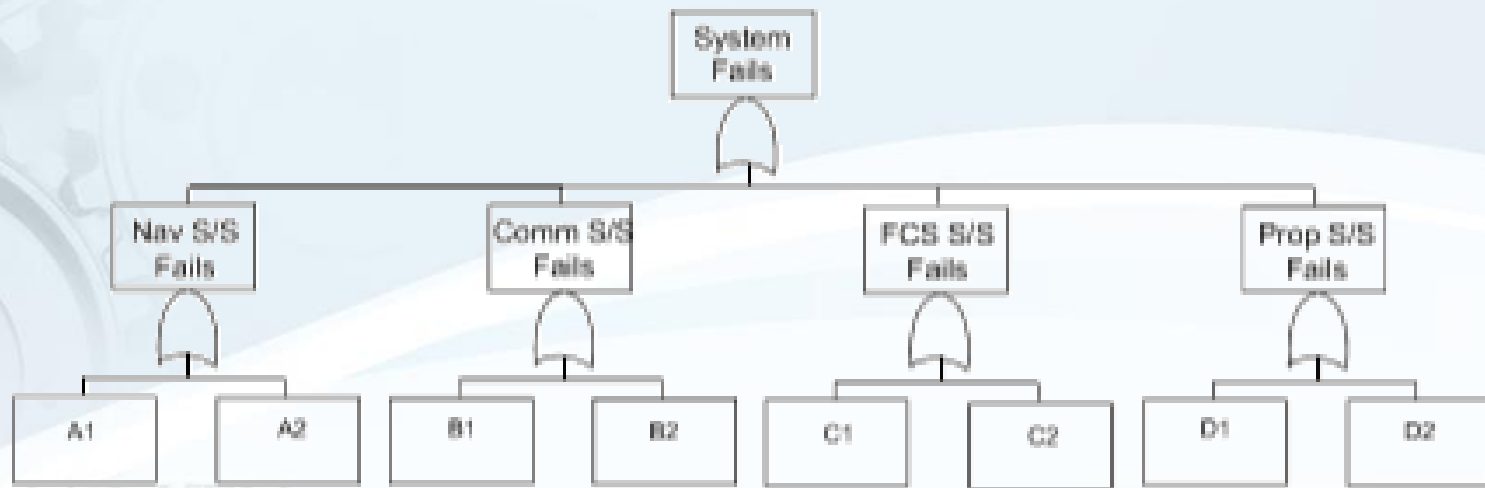
# Component Approach



- Immediate breakdown by component
- Ignores immediate cause-effect relationships
- Tends to logically overlook things for large systems

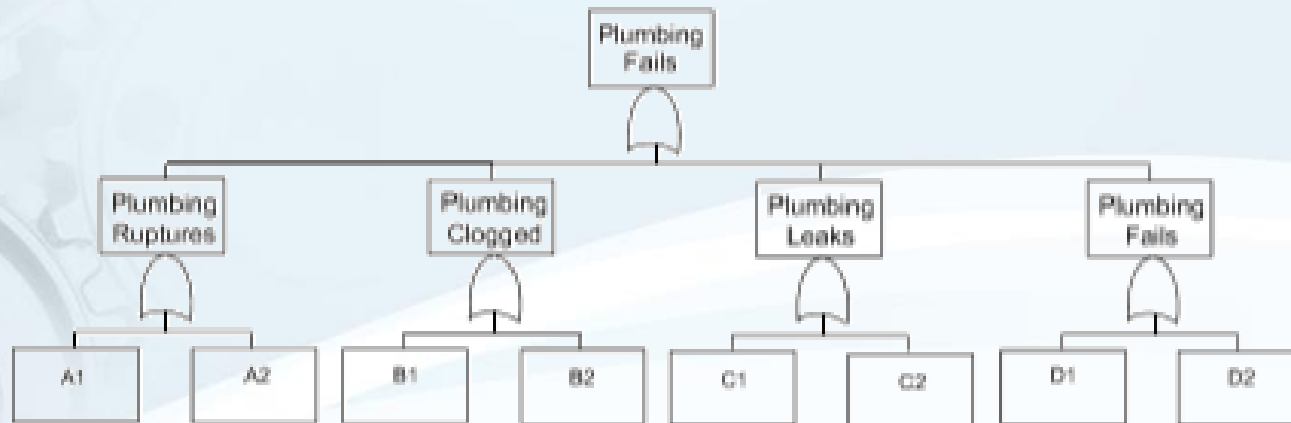
# Subsystem Approach

- Breakdown by subsystem
- Ignores immediate cause-effect relationships
- There can be hazard overlap between subsystems
- Tends to logically overlook things
- Eventually switch back to Functional approach



# Scenario Approach

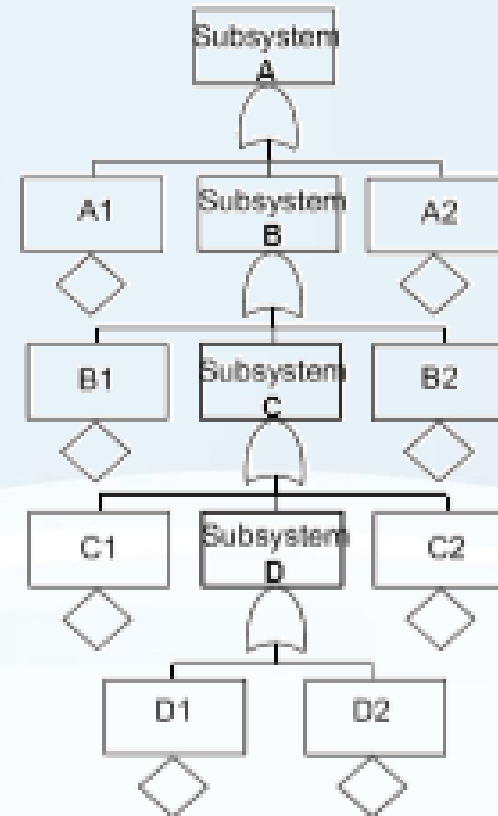
- Breakdown by Scenario
- Sometimes necessary to start large FTs
- Ignores immediate cause-effect relationship
- Eventually switch back to Functional approach
- Could be some overlap between subsystems



QUALITY

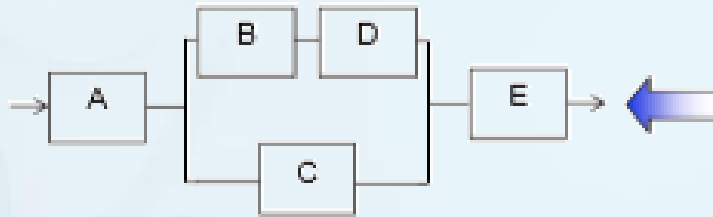
# Functional Approach

- Breakdown by system function
- FTA follows system function
- Follows logical cause-effect relationship
- Has more levels and is narrower
- Less prone to miss events
- More structured and complete analysis
- Use for about 90% of applications
- FTA follows functional command path
- Structured approach

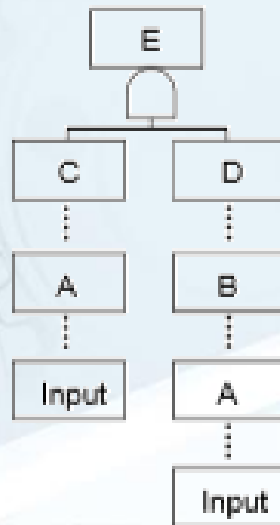


Recommended approach

# Functional Approach



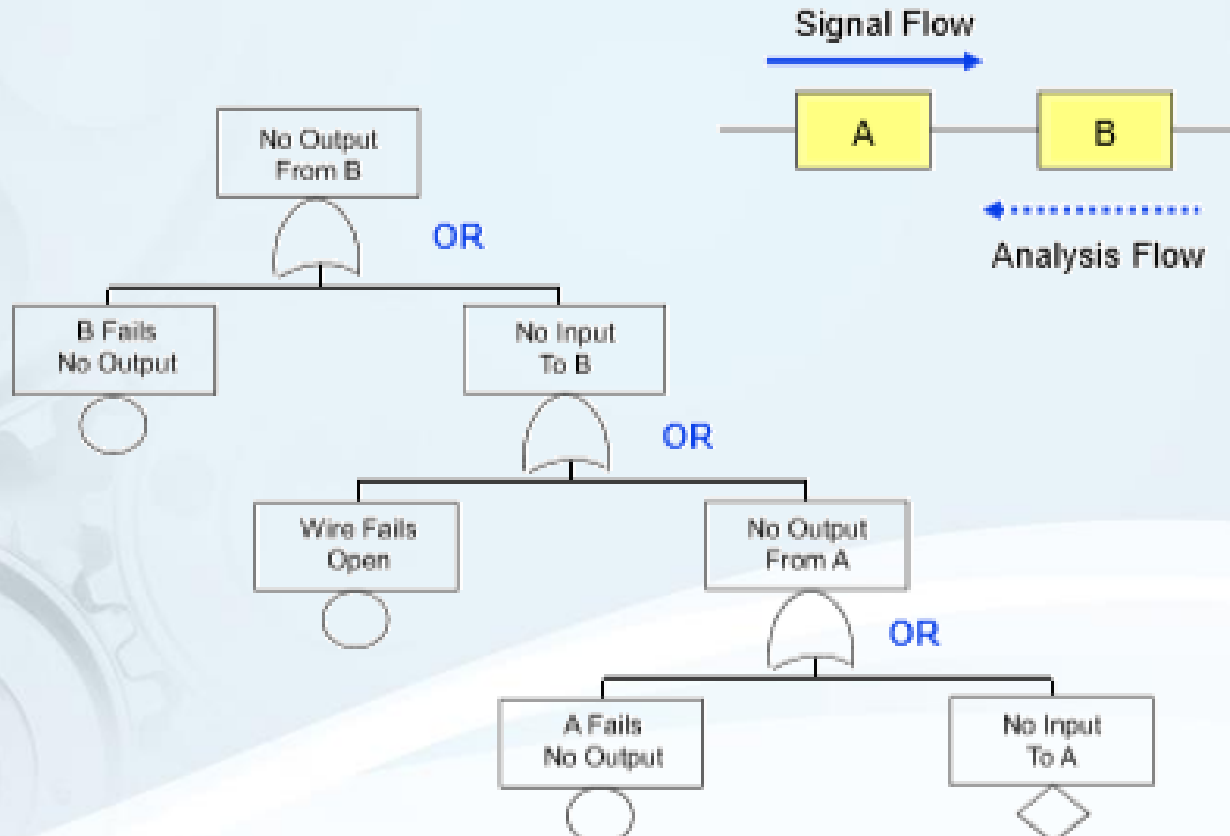
Follow the functional path



- Start at UE location (E in this example)
- Follow signal flow backwards
- Take each component one at a time

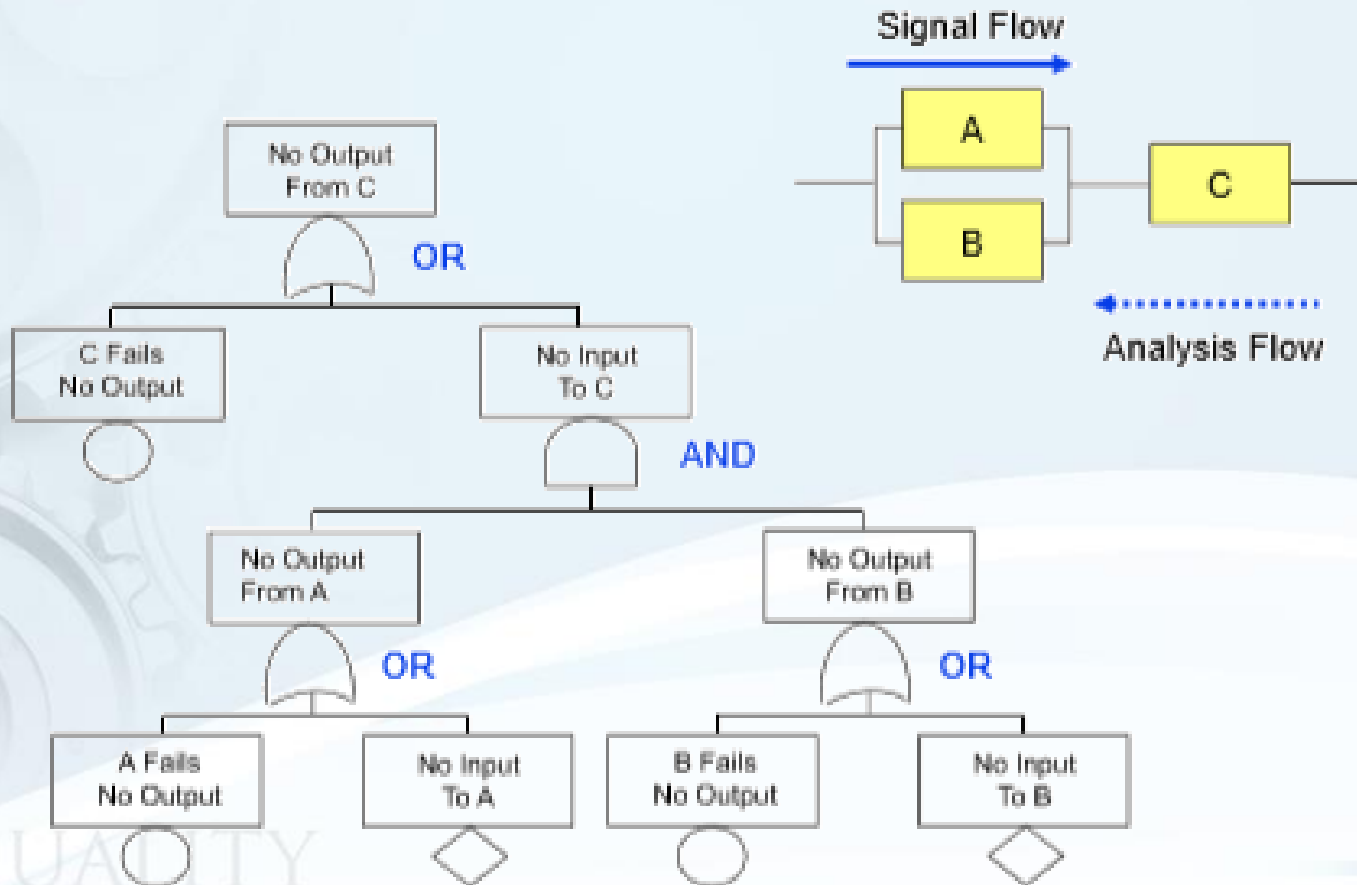
QUALITY

# Serial Example



QUALITY

# Serial – Parallel Example





# FTA construction methodology

- Construction at each gate involves a 3 step question process:
  - Step 1 – Immediate, Necessary and Sufficient (I-N-S) ?
  - Step 2 – Primary, Secondary and Command (P-S-C) ?
  - Step 3 – State of the Component or System (S-C/S) ?

These are the 3 key questions in FTA construction

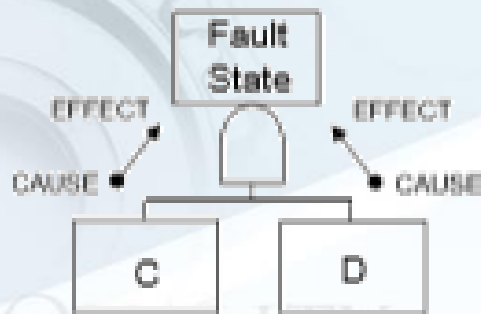
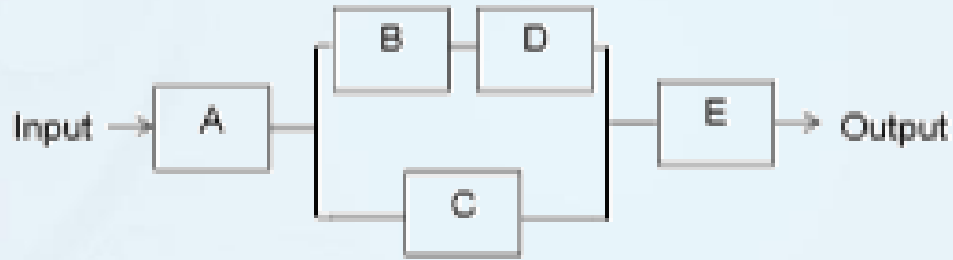
QUALITY

# Step 1

- Step 1 – *What is Immediate, Necessary and Sufficient (I-N-S) ?*
  - Read the gate event wording
  - Identify all *Immediate*, *Necessary* and *Sufficient* events to cause the Gate event
    - Immediate – do not skip past events
    - Necessary – include only what is actually necessary
    - Sufficient – do not include more than the minimum necessary
  - Structure the I-N-S casual events with appropriate logic
  - Mentally test the events and logic until satisfied

QUALITY

# Step 1



C and D are *Immediate*  
C and D are *Necessary*  
C and D are *Sufficient*. } To cause Fault of E

# Step 2

- Step 2 – *What is Primary, Secondary and Command (P-S-C) ?*
  - Read the gate event wording
  - Review I-N-S events from Step 1
  - Identify all **Primary**, **Secondary** and **Command** events causing the Gate event
    - Primary Fault – basic inherent component failure
    - Secondary Fault – failure caused by an external force
    - Command Fault – A fault state that is commanded by an upstream fault or failure
  - Structure the P-S-C casual events with appropriate logic

If there are P-S-C inputs, then it's an OR gate

# Primary, Secondary, Command Failure

- **Primary Failure**

- A *component failure* that cannot be further defined at a lower level.
- Example – diode inside a computer fails due to materiel flaw.

- **Secondary Failure**

- A component failure that can be further defined at a lower level, but is *not defined in detail* (ground rules).
- Example – computer fails (don't care about detail of why).
- A component failure that is *caused by an external force* to the system, can be further defined.
- Example – Fuel tank ruptures due to little boy shooting it with an armor piercing bow and arrow.
- They are also important when performing a Common Cause Analysis.

QUALITY

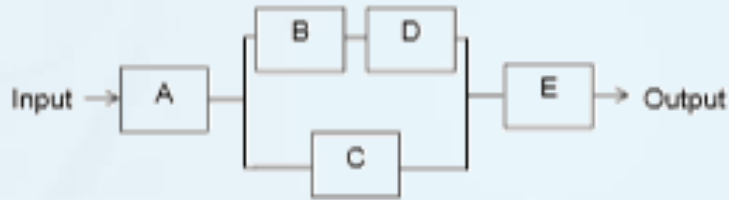
# Primary, Secondary, Command Failure

- **Command Failure**

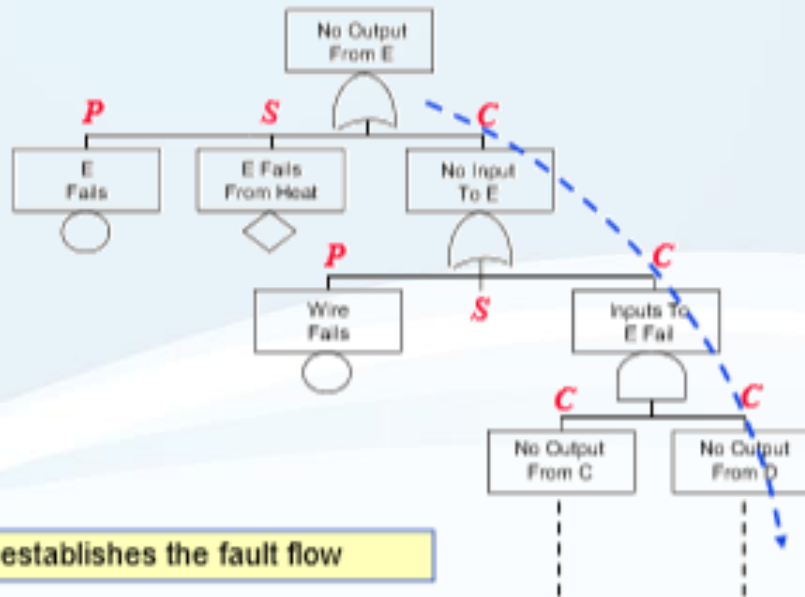
- A fault state that is commanded by an upstream fault / failure.
- Normal operation of a component, except in an inadvertent or untimely manner. The normal, but, undesired state of a component at a particular point in time.
- The component operates correctly, except at the wrong time, because it was commanded to do so by upstream faults.
- Example – a bridge opens (at an undesired time) because someone accidentally pushed the Bridge Open button.

QUALITY

# Step 2



P = Primary Failure  
S = Secondary Failure  
C = Command Failure



The Command path establishes the fault flow

QUALITY

# Step 3

- Step 3 – *Is it a State of the Component or System (S-C/S) fault ?*
  - Read the gate event wording
  - Identify if the Gate involves
    - ◆ a *State of the Component* fault
      - ☞ Being directly at the component level
      - ☞ Evaluating the causes of a component failure
    - ◆ a *State of the System* fault
      - ☞ Being a system level event
      - ☞ If it's not a state of the component fault
  - Structure the casual events with appropriate logic

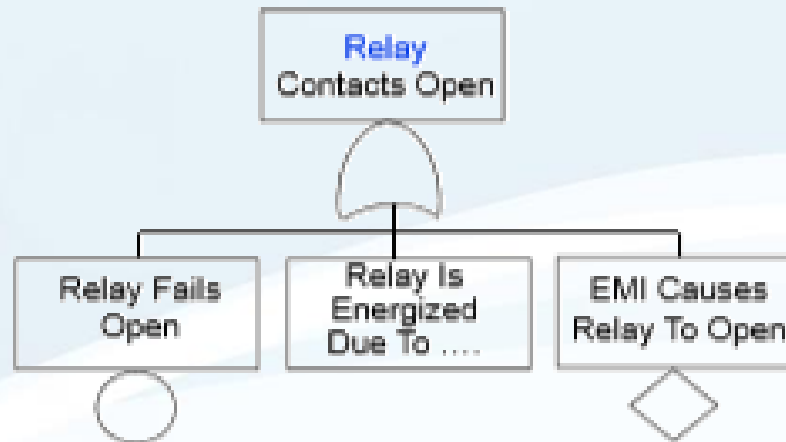
QUALITY

OSM\$\$N\$E\$X



## Step3 (Cont.)

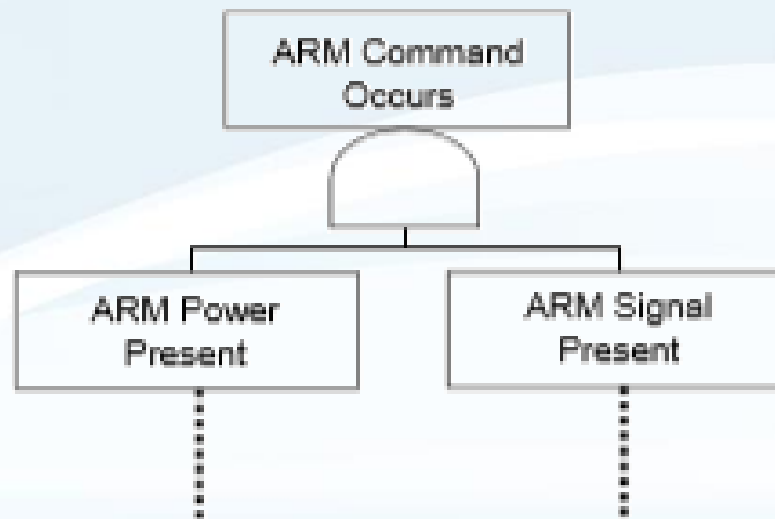
- If State of the **Component**, then:
  - Ask “what are the P-S-C causes”
  - Generally this results in an OR gate
  - If a Command event is not involved, then this branch path is complete



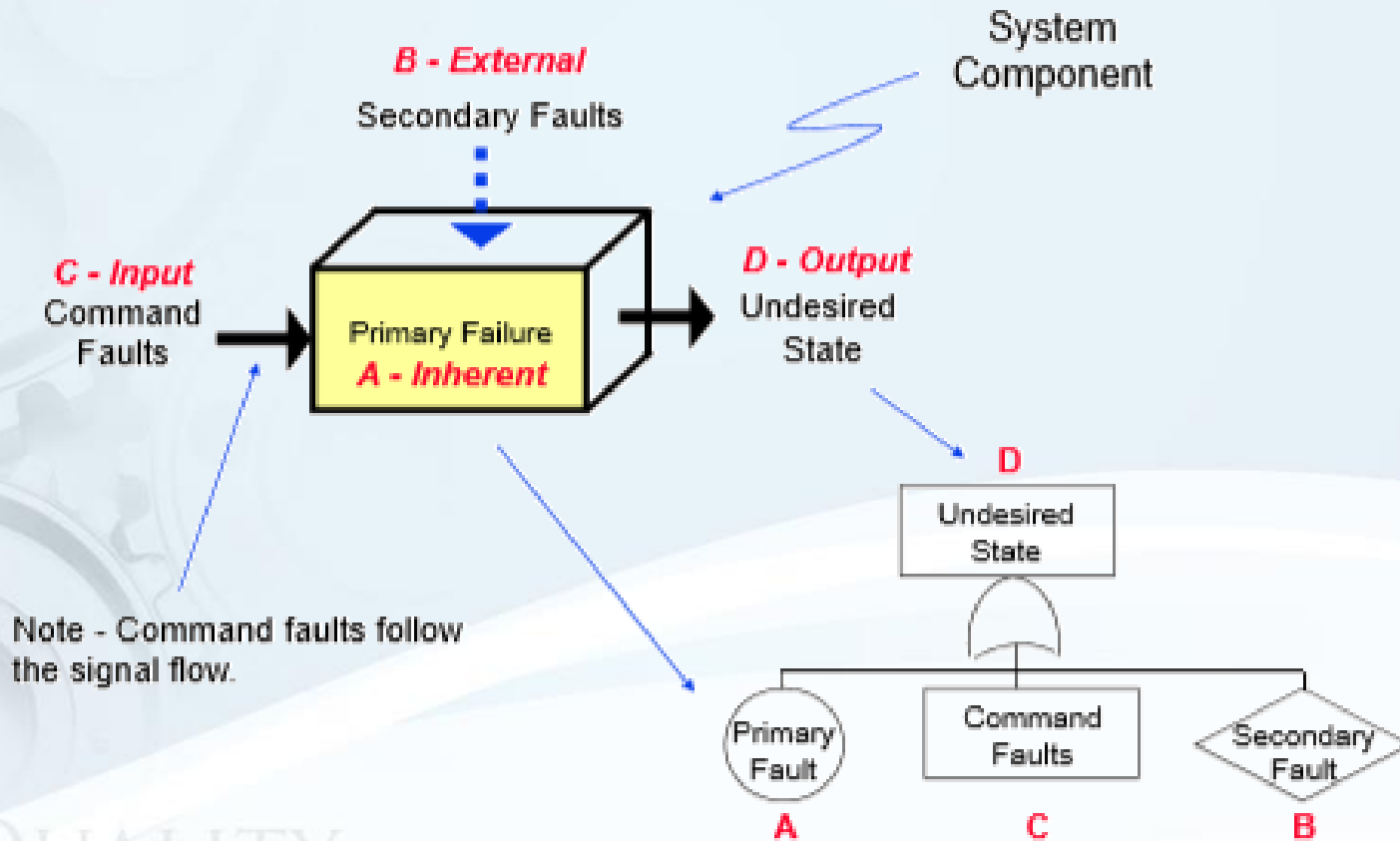
QUALITY

## Step3 (Cont.)

- If State of the **System**, then:
  - Ask "what is I-N-S" to cause event
  - Compose the input events and logic (functional relationships)
  - This gate can be any type of gate, depending on system design
  - The input events are generally gate events

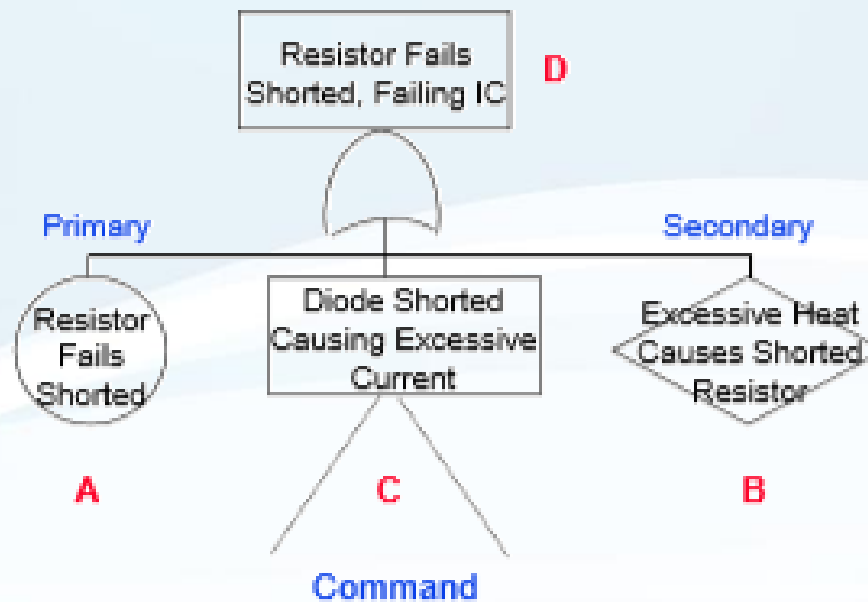
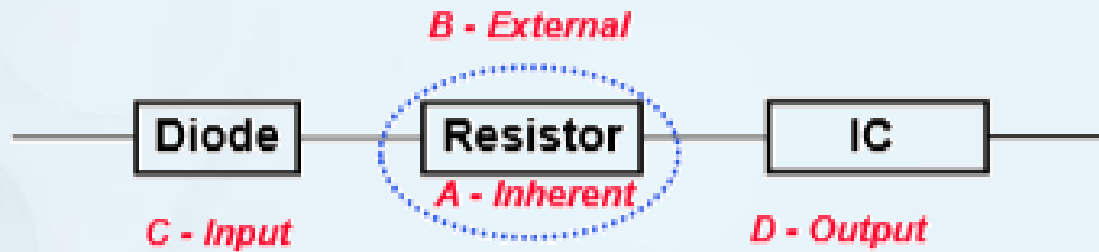


# P-S-C relationship with FTA



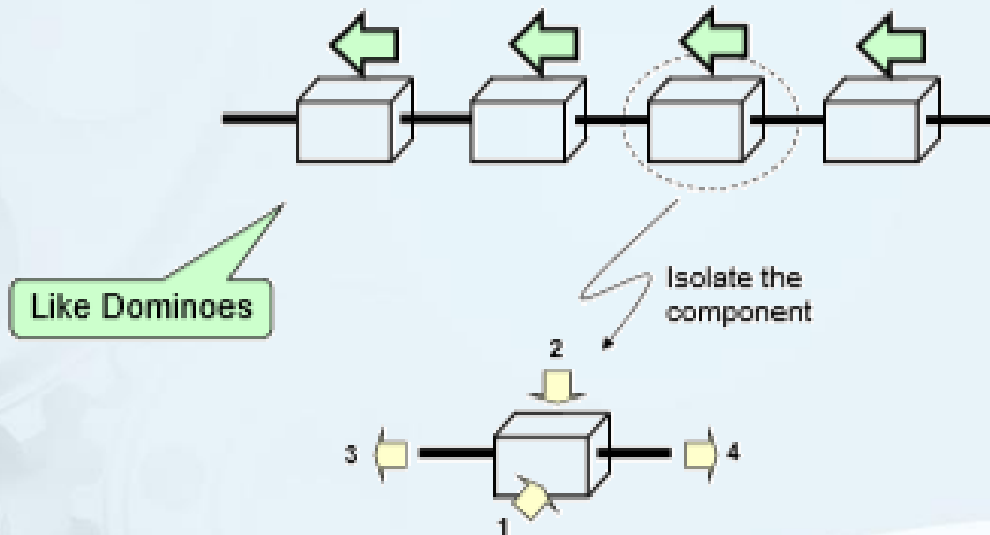
QUALITY

# P-S-C Example



QUALITY

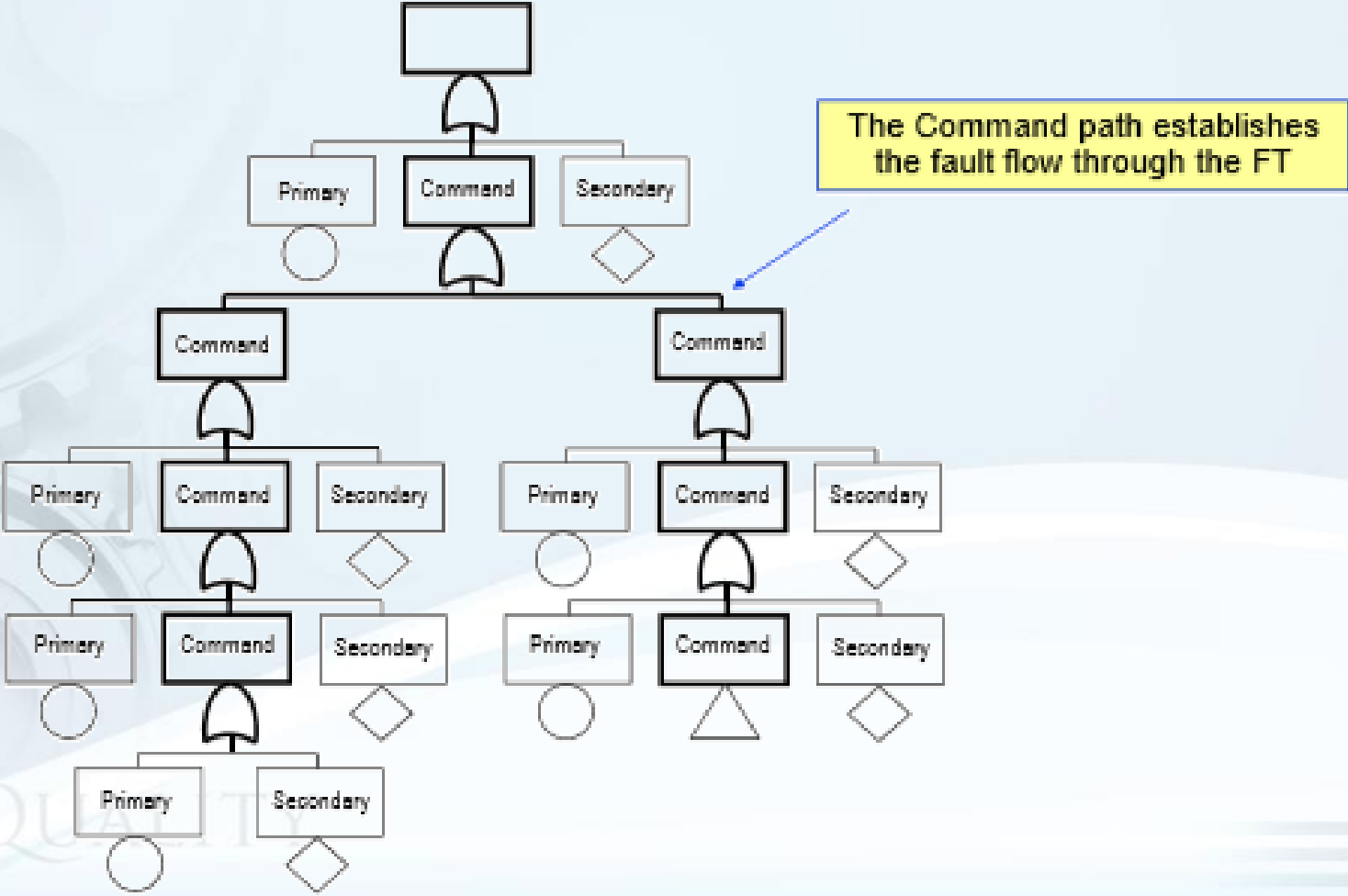
# Isolate and Analyse



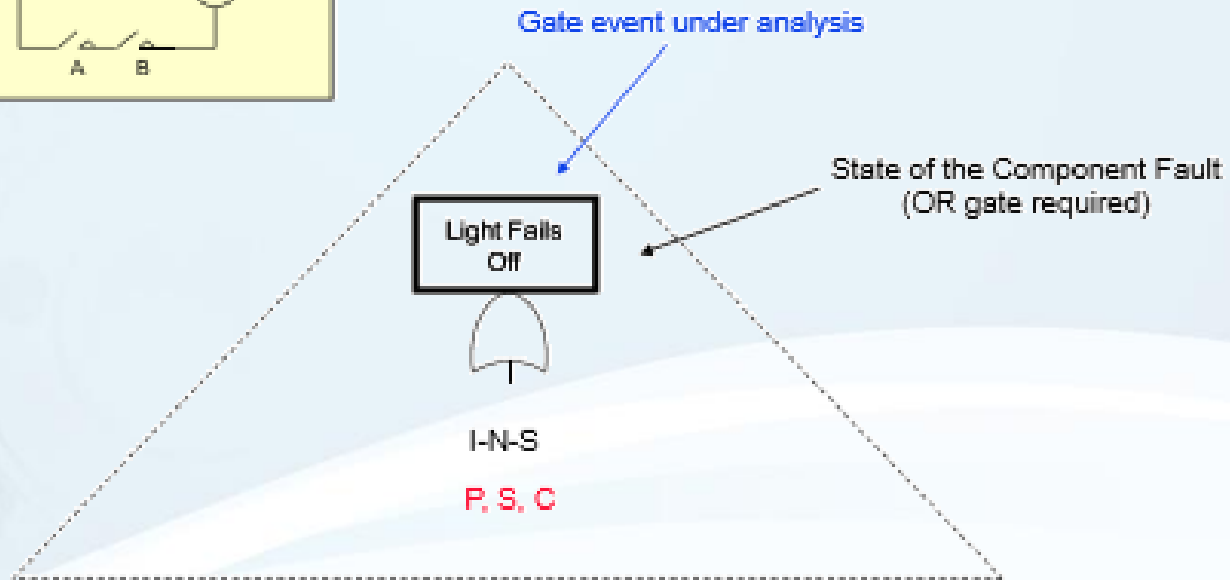
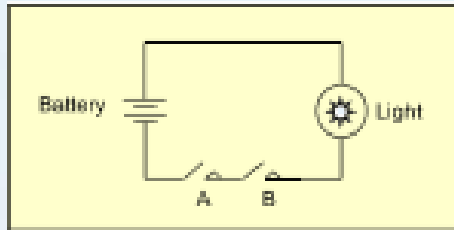
Analysis Views:

- 1) Primary - look inward
- 2) Secondary - look outward for incoming environmental concerns
- 3) Command - look backward at incoming signals
- 4) Output - look forward at possible undesired states that can be output

# Example of common path

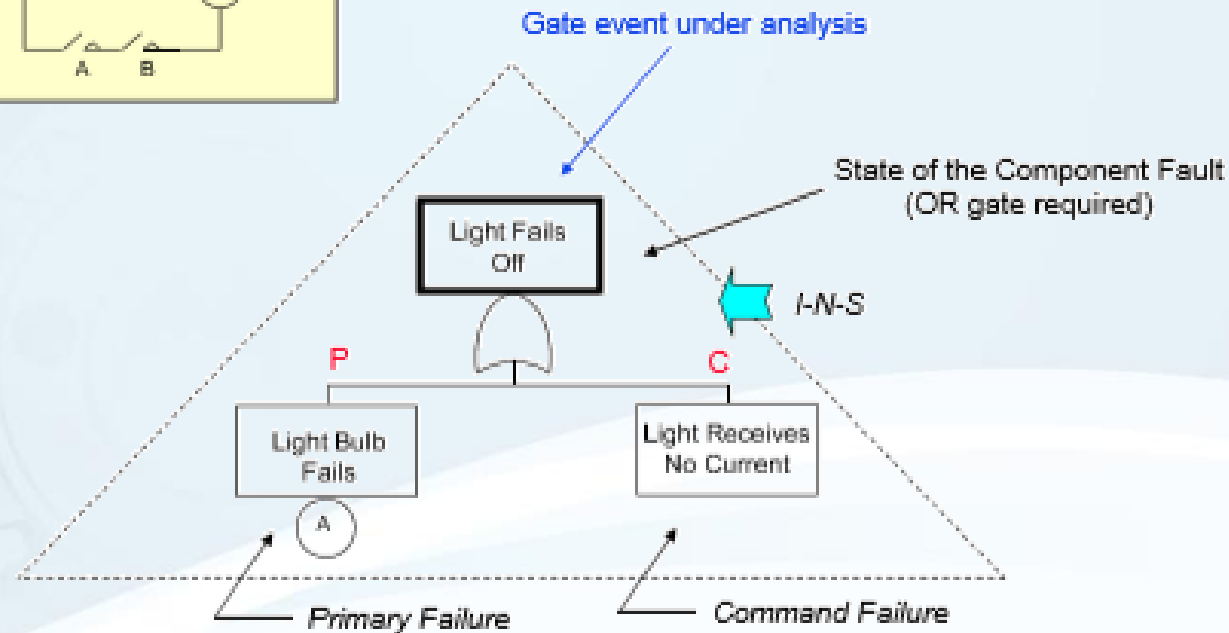
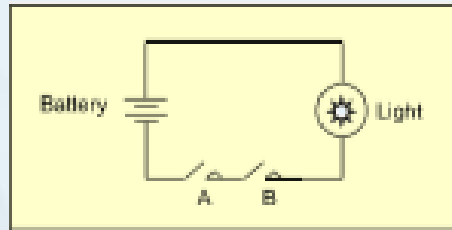


# Construction Example



P – primary failure  
S – secondary failure  
C – command fault

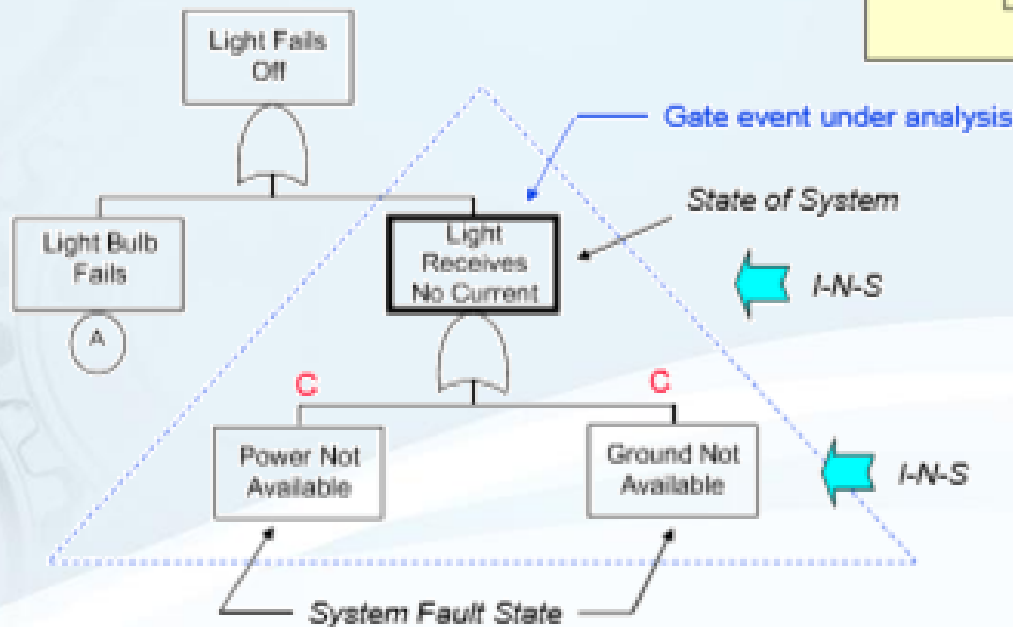
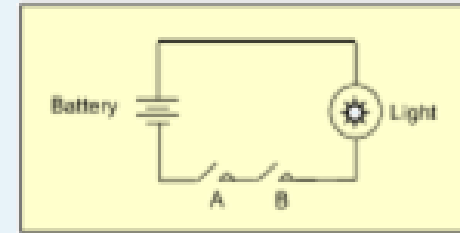
# Construction Example (Cont..)



Note – This uses P-S-C, I-N-S and S-C/S

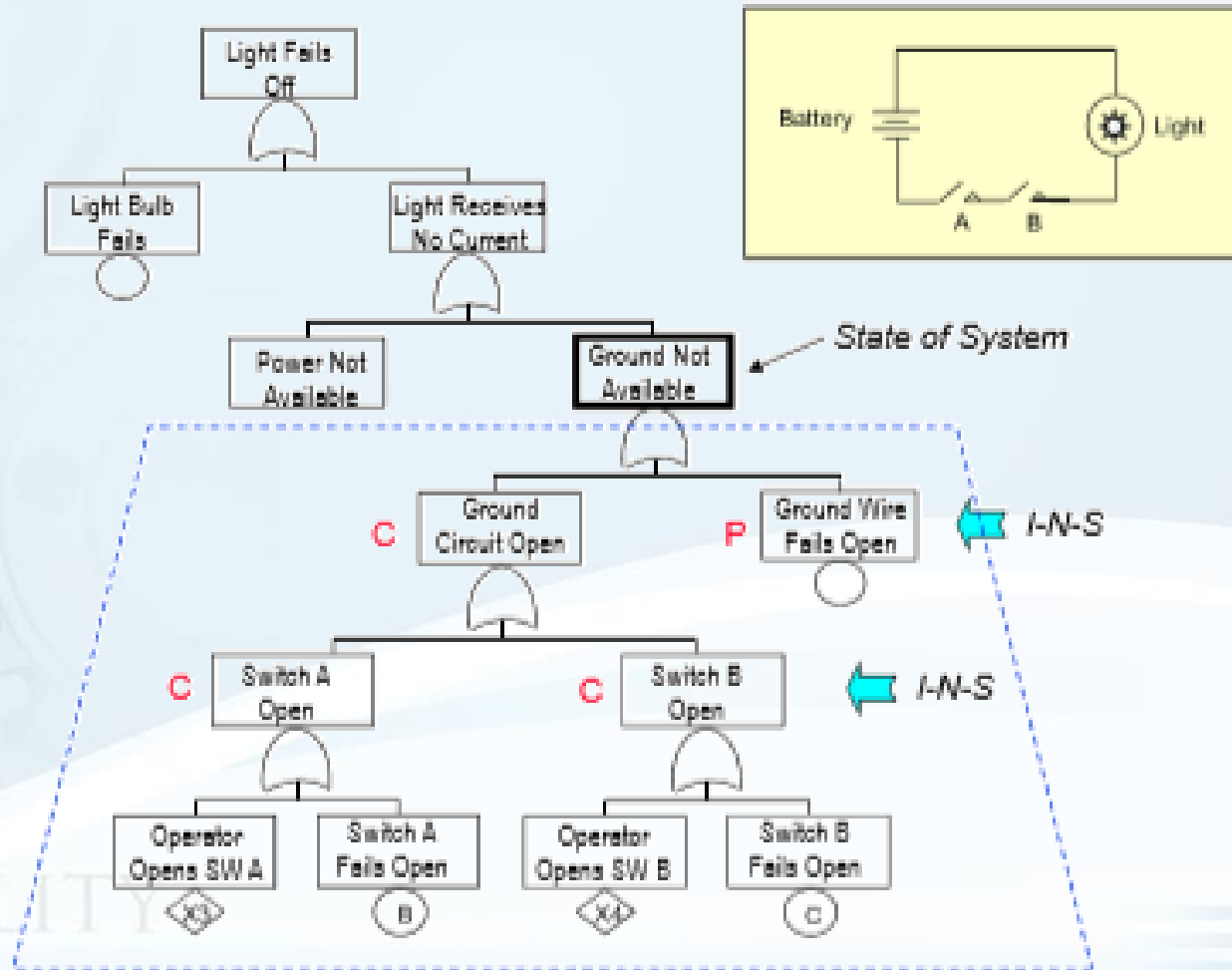


# Construction Example (Cont..)

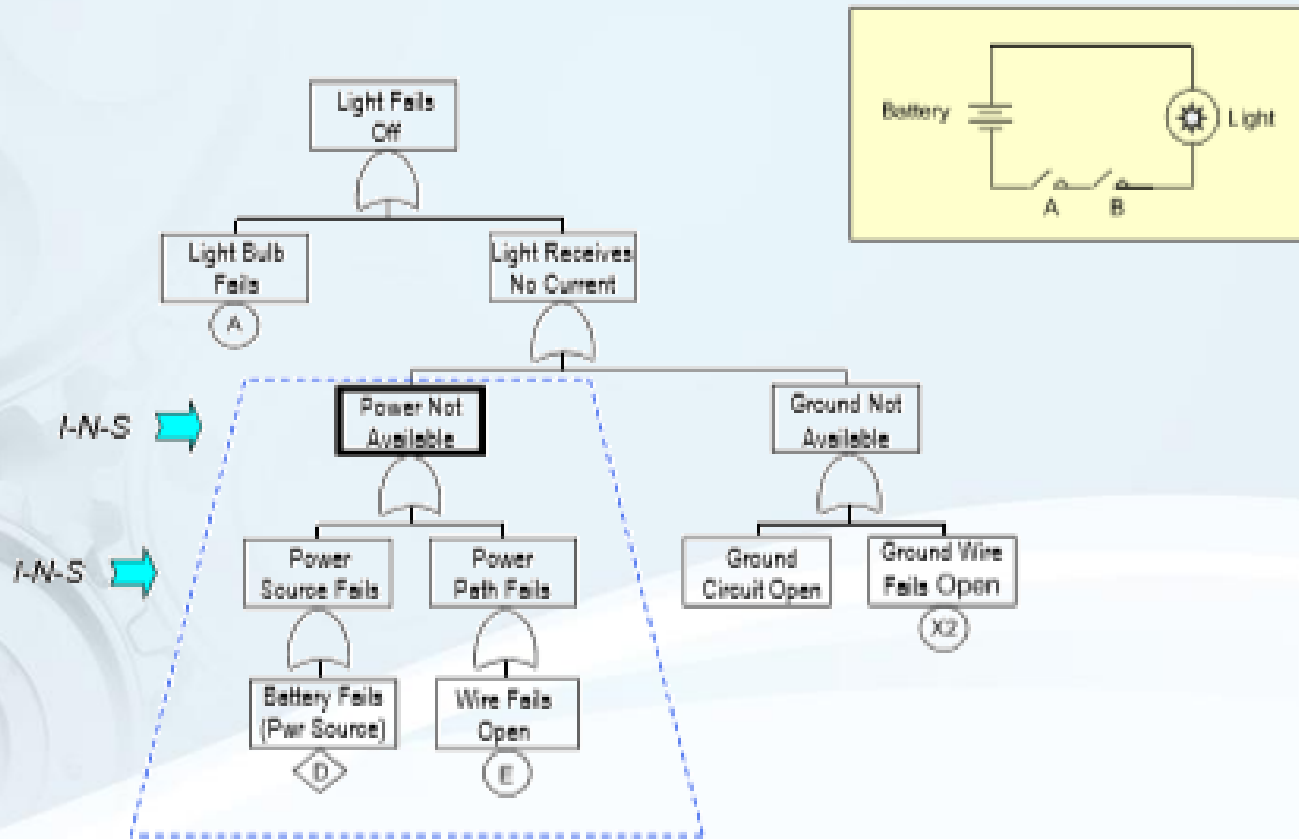


C - command fault

# Construction Example (Cont..)

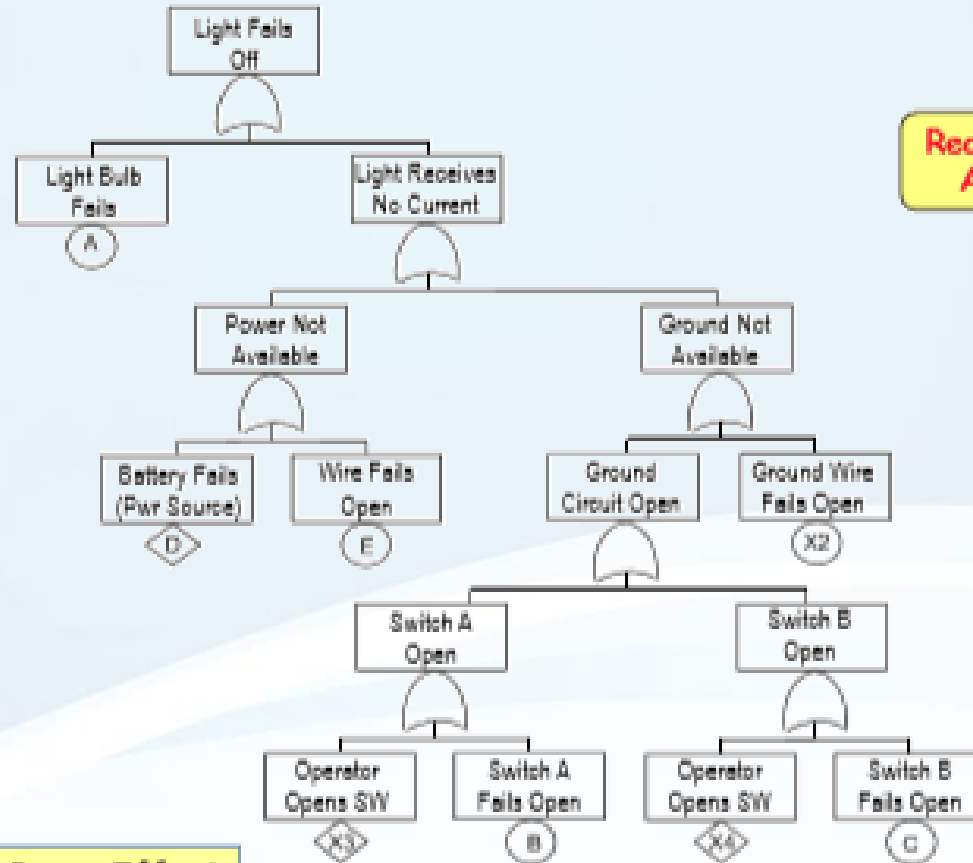


# Construction Example (Cont..)



QUALITY

# FTA Process – Functional Approach



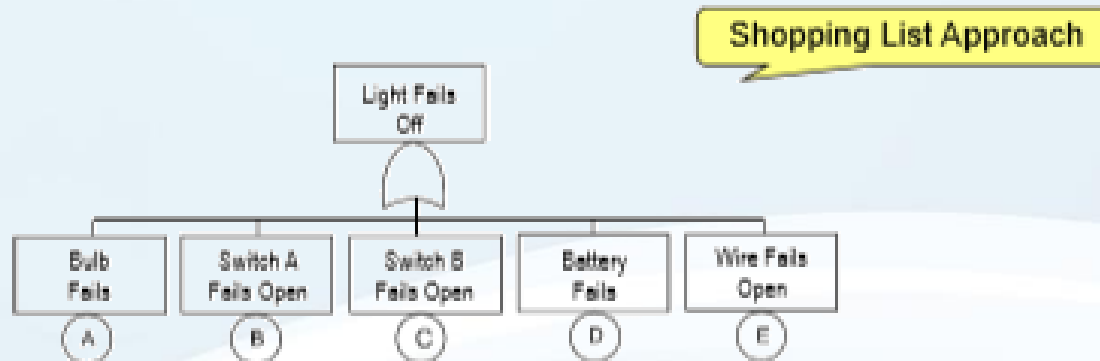
Recommended Approach

Note that logical Cause-Effect relationships are visible

10?

# FTA Process - Unstructured

- The unstructured approach jumps ahead
  - Misses some important items, such as the total number of wires involved, human interaction, etc.
  - Does not depict system fault logic



Note that Cause-Effect relationship is **not** visible

# FTA CONSTRUCTION RULES

QUALITY



# Rule #1

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

- Use the right tool
- Use the tool correctly
- Remember, 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
- Know when to use another tool

# Rule #2

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

- 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/reasonable model
  - don't solve the wrong problem
  - don't try the impossible
  - make sure analysis will meet desired objectives/goals



# Rule #3

- Rule #3 -- Establish Your FTA Ground Rules
  - 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

## *Rule #4 -- Intentionally Design Your Fault Tree*

- Follow FTA ground rules and formats
  - Make checks against ground rules
- Establish name convention for Events, MOEs and Transfers
  - use a methodology
  - by hardware type, supplier, subsystem
  - short names are usually better (long names becomes burdensome, time consuming)
- Maintain event databases and cross references
  - basic failure events, gate events, condition events, MOE's, transfers
- Establish tree structure approach
  - functional or subsystem

# Rule #4 (continued)

- Determine level of analysis detail
  - subsystem, LRU, component
- 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
- 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 FT programs and design around their capabilities

# Rule #4 (continued)

- Maintain tree metrics
  - event counts [?] Basic Events, Gate Events
  - complexity
  - complexity
- Tree size (more effort for larger trees)
  - small (< 100 event)
  - medium (100 to 750 events)
  - large (750 to 2,000 events)
  - huge (>2,000 events)
- Conduct tree peer review
  - other FT experts
  - system designers

# Rule #5

## *Rule #5 -- Know Your System*

- 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, maintenance procedures
- 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
  - reviews - peer , designers, customer
- The model and design data can be iterative
  - preliminary model progresses to detailed model

# Rule #6

## *Rule #6 -- Understand Your Failure Data*

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

QUALITY

# Rule #7

## *Rule #7 -- Know Your Fault Tree Tools*

- 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)

# Rule #7 (continued)

- Know tool limits
  - tree size
  - cut set size
  - plot size
- Understand cutoff methods, some can cause errors
- Gate probabilities could be incorrect when MOE's are involved

QUALITY



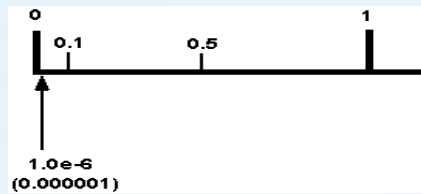
# Rule #8

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

- Failure rates and probabilities are between 0 and 1
- FT's generally deal with small numbers ( $< 1.0e-6$ )
- Small numbers are somewhat abstract
- The exponent size is of prime interest ( $e-6$ ,  $e-15$ ,  $e-35$ )
  - Decimal places are somewhat significant within the same range ( $1.11e-6$  vs  $1.97e-6$ )
  - Decimal places are not as significant for a wide range ( $1.1e-6$  vs.  $1.778e-9$ )
  - As numbers get very very small, decimal place are probably insignificant (ie,  $1.0e-35$  vs.  $1.2e-35$ )

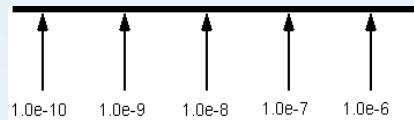
QUALITY

# Rule #8 (continued)



Probability range is between 0 and 1. FT events and cut sets also fall into this probability range. A number of  $1.0e-6$  looks very abstract on this chart.

1 failure per million hrs =  $0.000001 = 1.0e-6$



Looking at a small number within a range of small numbers provides more valuable information.

What Are Small Numbers ?

# Rule #8 (continued)

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

- Don't get carried away with numbers
  - All results are essentially estimates for relative comparisons
  - is system  $1.0e-3$  or  $1.0e-7$  is relevant
  - is system  $1.1e-6$  or  $8.7e-6$  is not as relevant
  - is system  $1.1e-6$  or  $1.123767e-6$  is not relevant
- Remember, the model is *only* a model and does not have 100% fidelity to the true system, therefore, everything is somewhat relative

QUALITY

# Rule #9

## *Rule #9 -- Understand Your Results*

- Make reasonableness tests on the results
  - are the results correct
  - look for analysis errors (data, model, computer results)
  - are CS's credible and relevant, if not revise tree
  - take nothing for granted from the computer
  - test your results via hand calculations
- Verify that the FTA goals were achieved
  - are the results meaningful
  - was the analysis objective achieved
  - was the right tool used

QUALITY

# Rule #9 (continued)

- 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
- Effect of MOEs is very important
  - they can cause large numerical impact or none at all
  - review carefully

QUALITY

# Rule #10

## *Rule #10 – Remember FT's Are Models*

- Remember that FT's are models
  - perception or model of reality
  - not 100% fidelity to exact truth
- Remember that models are approximations (generally)
  - not necessarily 100% exact
  - still a valuable predictor
  - Newton's law of gravity is an approximation
- Do not represent FTA results as an exact answer
  - use engineering judgment
  - small number are relative ( $2.0 \times 10^{-8}$  is as good as  $1.742135 \times 10^{-8}$ )
  - anything overlooked by the FTA skews the answer
    - ☞ minor things left out can make results conservative (understate results)
    - ☞ major things left out can be significant (overstate results)

# Rule #11

*Rule #11 -- Publish/Document Your Analysis And Results Completely*

- Formally document and publish the entire FTA
  - may need to provide to customer (product)
  - may need to defend at a later date
  - may need to modify at a later date
  - may perform a similar analysis at a later date
  - may need records for an accident/incident investigation
- Even a small analysis should be documented for posterity

QUALITY

# Rule #11 (continued)

- Provide complete documentation
  - problem statement
  - definitions
  - ground rules
  - references
  - comprehensive system description
  - data and sources (drawings, failure rates, etc.)
  - FT diagrams
  - tree metrics
  - FT computer tool description
  - results
  - conclusions



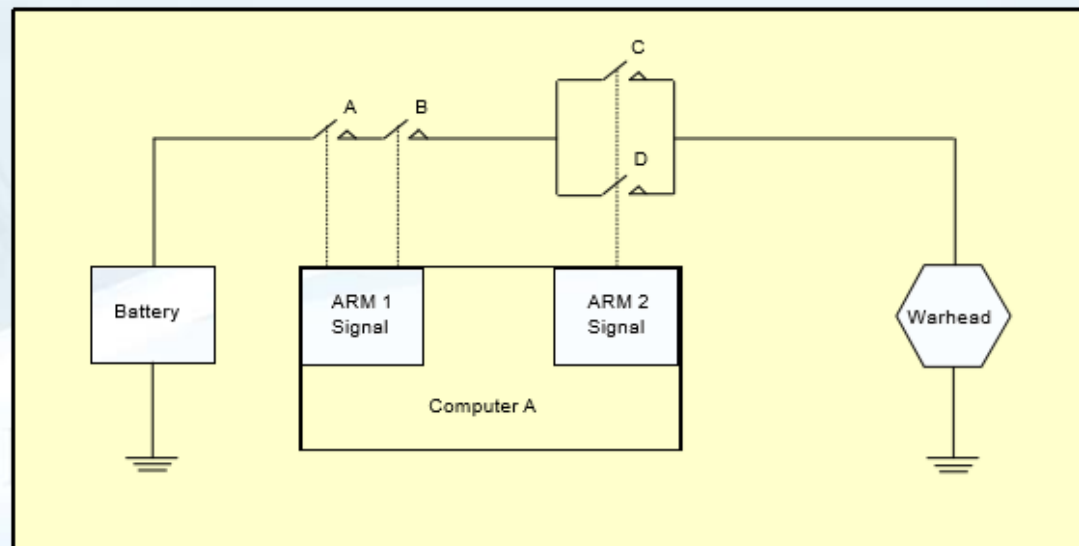
# FTA EXAMPLE

QUALITY



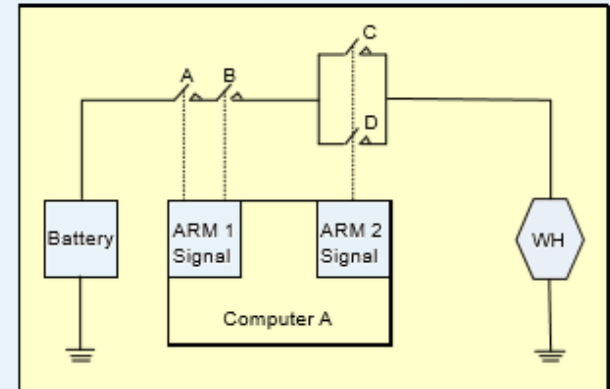
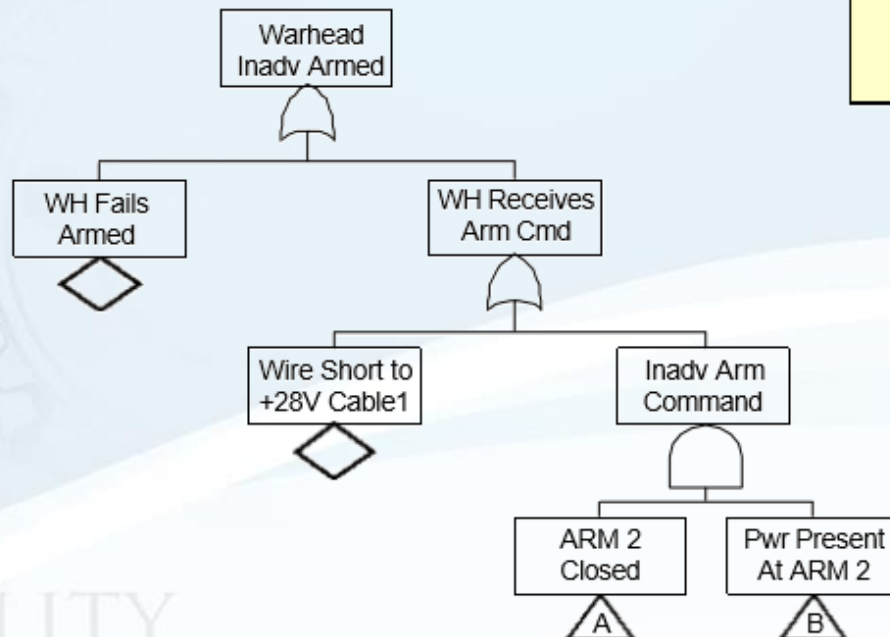
# FTA Example

- Construct a FT for the following system
  - The Undesired Event is “Inadvertent Warhead Arming”
  - Construct the Fault Tree
  - Ground Rules:
    - ☞ When all the switches are closed the Warhead receives the Arm command.

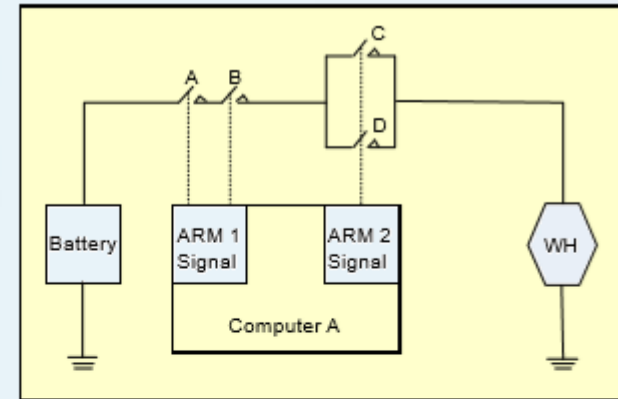
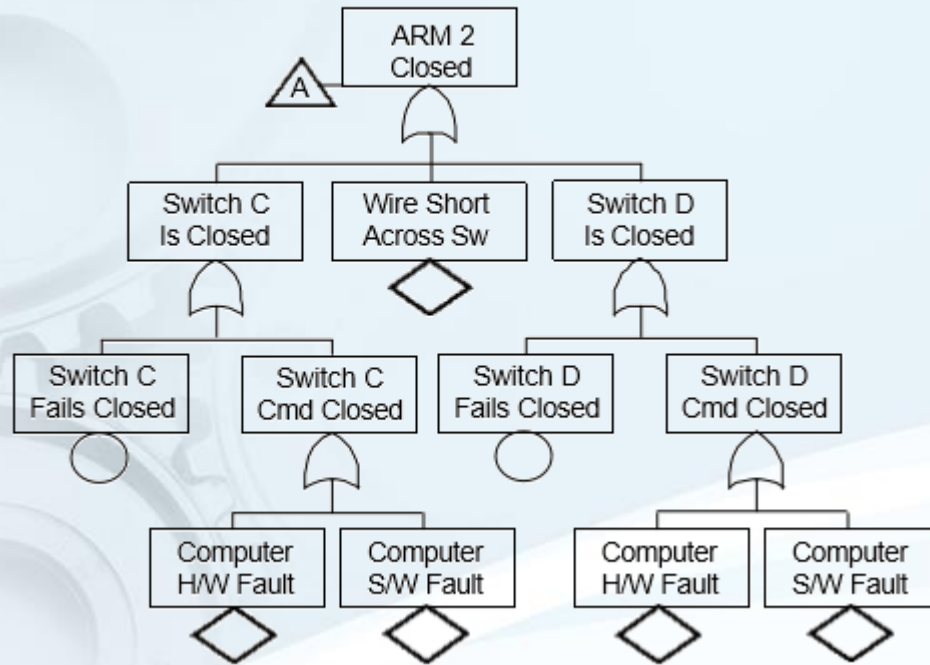


# FTA Example

Method 1 – Structured  
(Using Functional Approach)

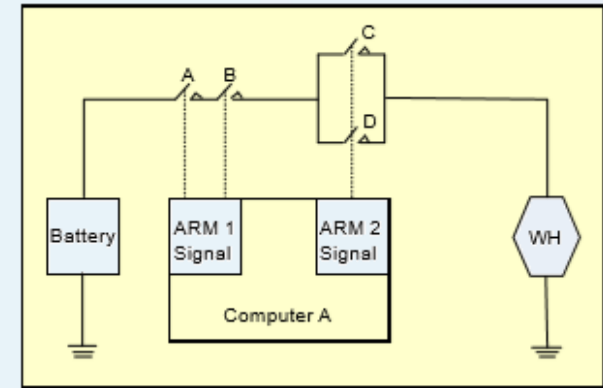
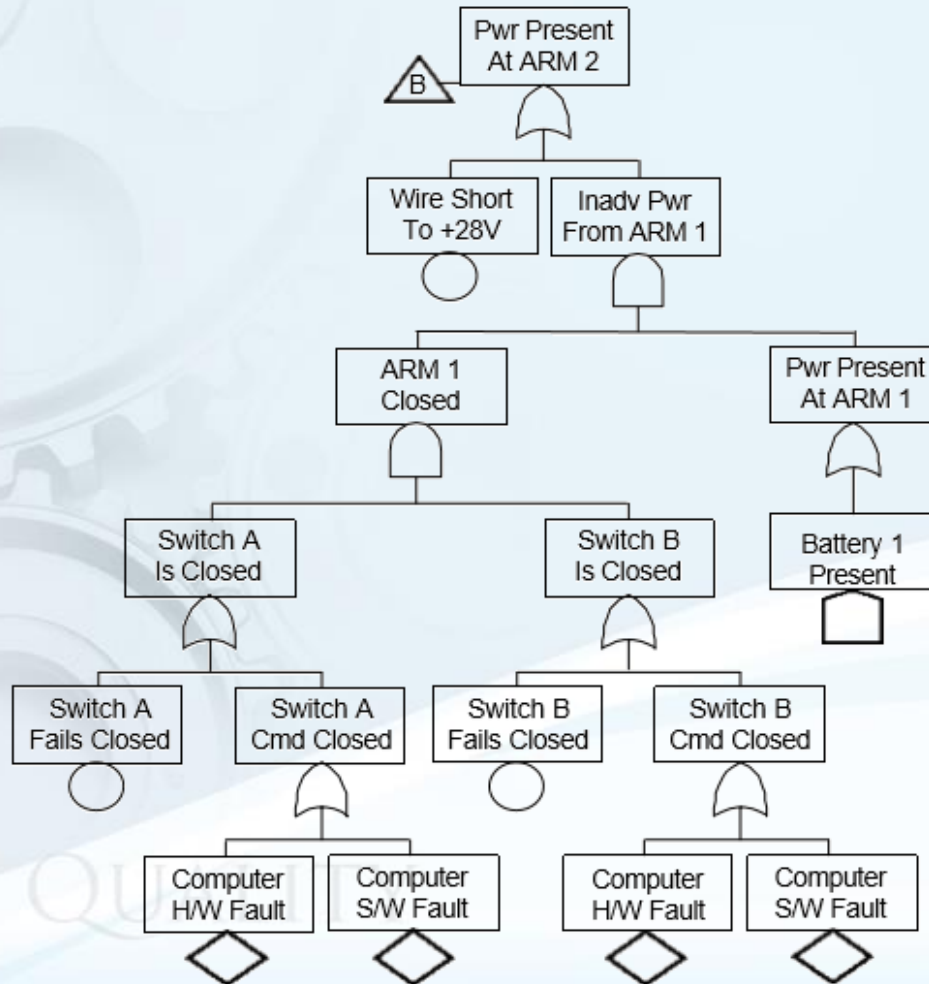


# FTA Example



QUALITY

# FTA Example



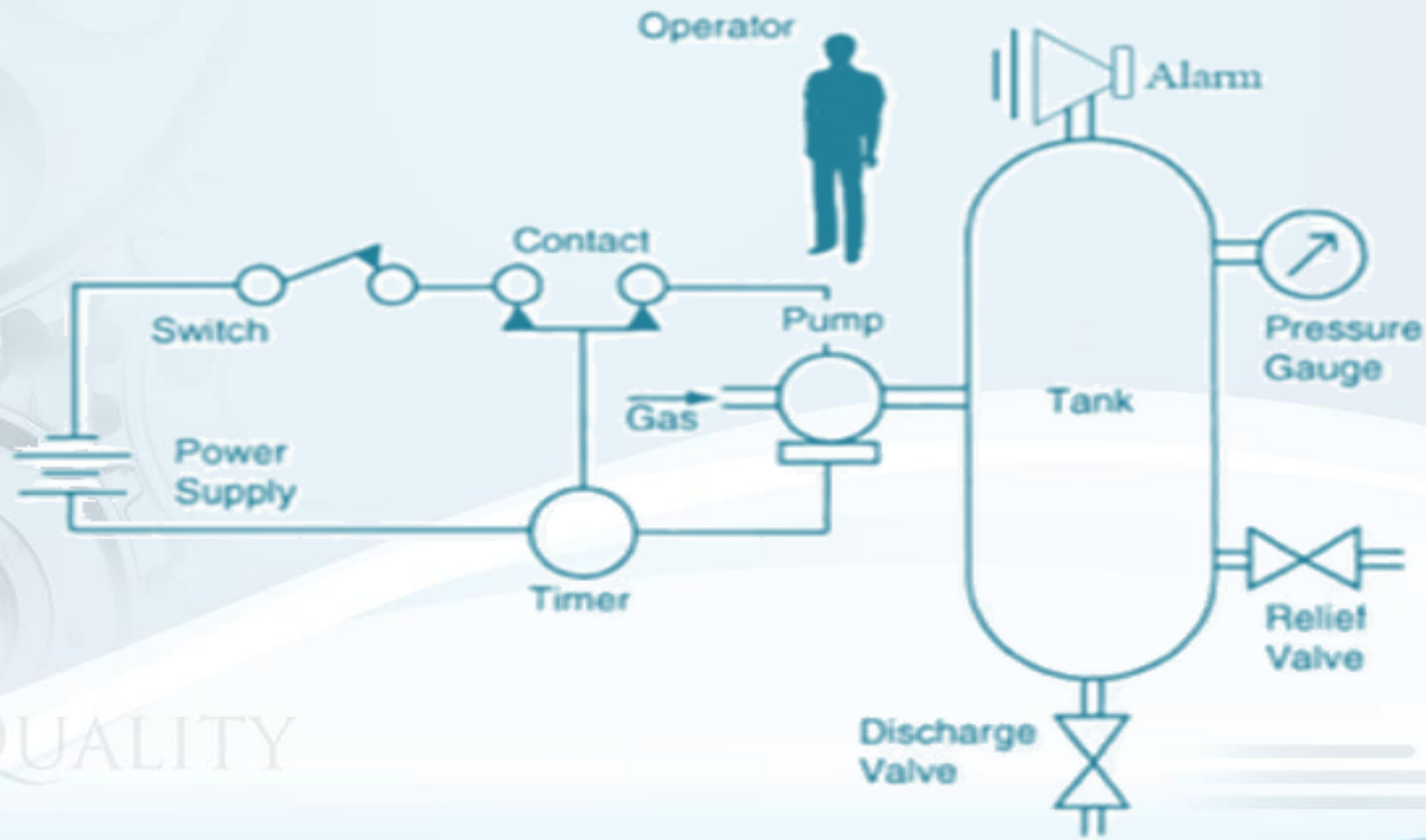
# Breakout Exercise 1

Create a the Fault Tree

QUALITY



# Pressure tank system (Undesired Event - Tank Rupture)



QUALITY



# Breakout Exercise 1: Develop a Fault Tree

The system shown in the figure discharges gas from a reservoir in to a pressure tank. The switch is normally closed and the pumping cycle is initiated by a operator who manually reset the timer. The timer contact closes and pumping starts. Well before any over pressure condition exists the timer times out and the timer contacts open. Current to the pump cuts off and pumping ceases. (to prevent tank rupture due to over pressure).

If the timer contact does not open, the operator is instructed to observe the pressure gauge and to open the manual switch, thus causing the pump to stop. Even if the timer and operator both fail, the overpressure can be relieved by relief valve. After each cycle, the compressed gas is discharged by opening the valve and then closing it before the next cycle begins.

At the end of the operating cycle, the operator is instructed to verify the operability of pressure gauge by observing the decreasing in the tank pressure as the discharged valve is opened. To simplify the analysis, we assume that the tank is depressurized before the cycle begin. The pressure gauge may fail during the new cycle even if its operability was correctly checked by operator at the end of last cycle. The gauge can fail before a new cycle if the operator commits an inspection error.



# Breakout Exercise 1: Create the Fault Tree

## Instructions

- Create the Fault Tree analysis for the identified hazard (Tank Rupture).
- Damping force low (In suspension system)
- AC not cooling.
- Axle welding crack. (Chassis system)
- Unintended deployment of air bag.
- Seat belt failure.
- Failure of Electrical control Unit.
- Use the flip chart for the exercise.
- Be prepared to present your team's to the class; rotate the team spokesperson.



**60 Minutes**

# AN INTEGRATED FMEA-FTA MODEL

- An integrated FTA and FMEA model is proposed for risk analysis of critical systems.
- Minimal cut sets derived from the fault trees are weighted based on Birnbaum's measure of importance and then the weights are used to revise Risk Priority Numbers (RPNs) obtained from the use of traditional FMEA techniques.
- Significant differences are revealed in risk rankings when the results from the hybrid approach are compared with those obtained from the classical risk analysis methods.
- Integrated FTA and FMEA has been employed coherently and concurrently to enhance and complement each other in several reliability applications.
- Forward integration (i.e., FMEA to FTA) and backward integration (i.e., FTA to FMEA) have been proposed.

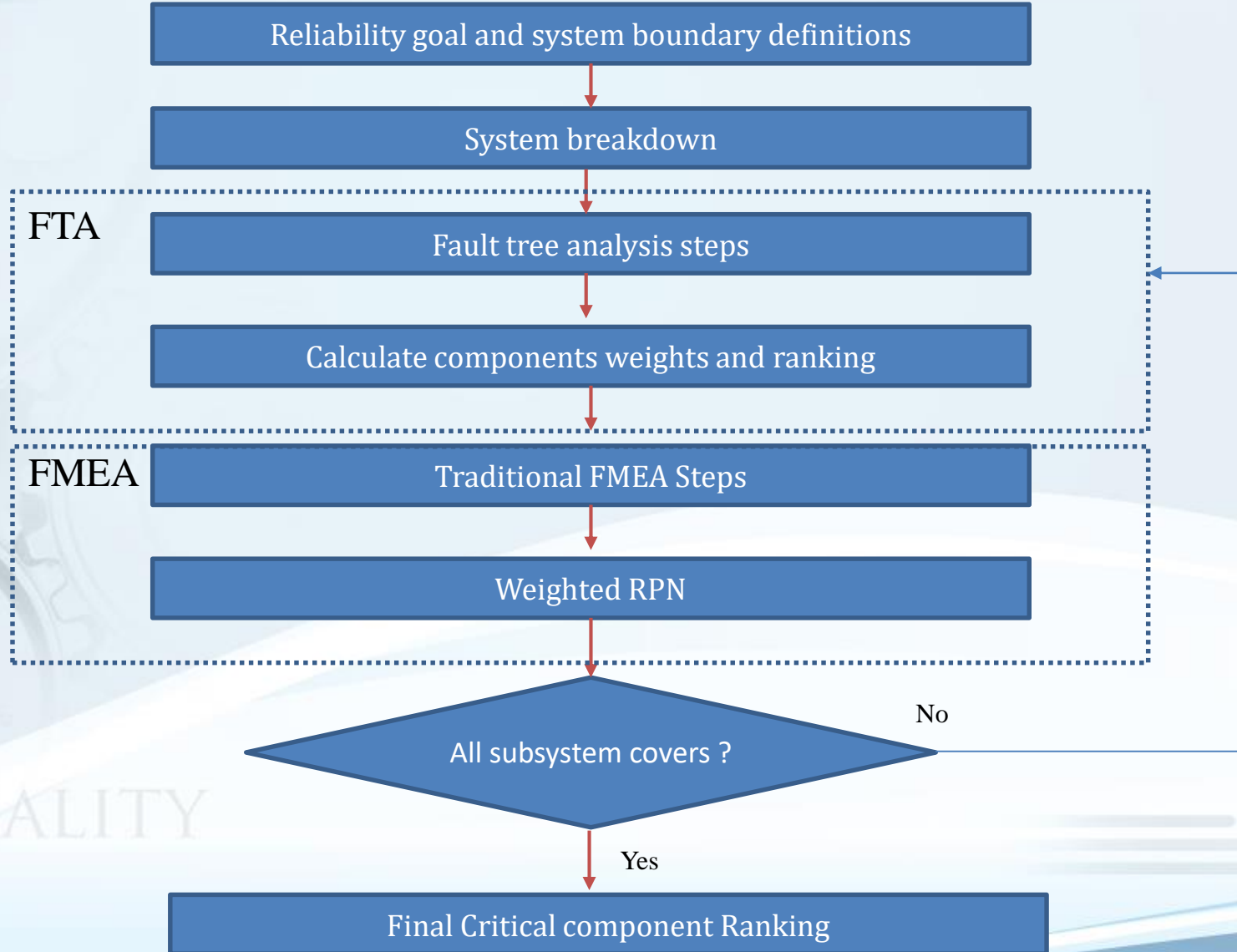
QUALITY

# AN INTEGRATED FMEA-FTA MODEL

- Applying both the FTA and FMEA techniques either simultaneously or in succession has been proven to be complementary and effective.
- Employed systematically, an integrated FTA-FMEA technique can provide a thorough evaluation of system safety concerns.
- FTA yields a comprehensive breakdown of faults leading to the undesired top event, whilst FMEA furnishes the exact fashion in which these faults exists and their direct effects on the top event, making the combination appropriate for failure and reliability analyses.
- Employing the combination of FTA and FEMA for identifying critical components and reliability analysis of highly complex systems is now popular among the analyst.
- In order to tap maximum benefits of an FTA-FMEA integration, minimal cut set theory will be used.

QUALITY

# Methodology



# Reliability Goal and System Boundary Definition

As a preliminary step in reliability assessment, particularly for complex systems, establishing generic reliability requirements of the system under consideration acts as a reference for further verification and validation. More so, the scope and element boundaries of the system require elucidation as complexity by definition may depict considerably large systems, interacting with several other elements.

## System Breakdown

It is nearly impossible to model a complex system using the traditional reliability analysis methods. The logical approach for this purpose is to subdivide the system into smaller units and employ probabilistic techniques to calculate overall reliability, based on reliability of the subsystems. Once the scope and boundaries are defined, clarity on further categorization of the system under consideration is facilitated, particularly for any expert or analyst with considerable knowledge.

## FTA Steps

- This phase includes slight modifications to the traditional FTA steps. The top event is defined and all immediate causes are identified. Again, secondary level events are specified and all root causes down to basic level are identified.
- The fault tree diagram is built and different fault combinations leading to top event are presented. At this stage, several fault trees may become necessary, dependent on the complexity of the system under consideration.
- Finally, minimal cut sets are obtained and their importance weights are evaluated.
- Assuming  $w$  is an independent function variable representing the importance of  $i$ th minimal cut set in the fault tree structure, we have:  $w_i = F(X_i)$ ,  $i = 1, 2$
- Where  $X_i$  represents the importance of the  $i$ th minimal cut set and  $F(X_i)$  is a function with independent variable  $X_i$ . By substituting Equation (3) in Equation (2), we have:  $WRPN_i = F(X_i) S_i O_i D_i$ ,  $i = 1, 2$
- Quantitative perspectives on dominant contributors to the top event can be provided by calculating the importance measure of the components in the system.



## FMEA Steps

- The additional tasks that should be implemented at this stage include revising traditional RPN values and ranking components based on the weighted RPNs.
- Experts brainstorm and report the results as in the traditional FMEA process.
- In this case, the minimal cut sets that were obtained from the fault trees aid the failure mode identification process.
- The weights are multiplied with RPNs obtained from the traditional FMEA procedure.
- It must be noted that minimal cut sets may include one or more components which should be assigned relative importance, as multiple failures are not considered.
- Components with highest RPNs may not necessarily possess highest WRPNs in this methodology, hence added criteria for ranking.
- An integrated FTA-FMEA worksheet carrying all necessary data from the traditional FTA and ending with suggested preventive actions.

# An Integrated FTA-FMEA Worksheet

System: Component: Team:				Report No. Prepared by: Date:											
FTA				FMEA											
MCS	Component/ subsystem	$X_i$	$F(X_i)$	Fail. mode	Fail. cause	Fail. effect	Ctrl. Mech.	S	O	D	RPN	Priority	WRPN	New priority	Action required

QUALITY



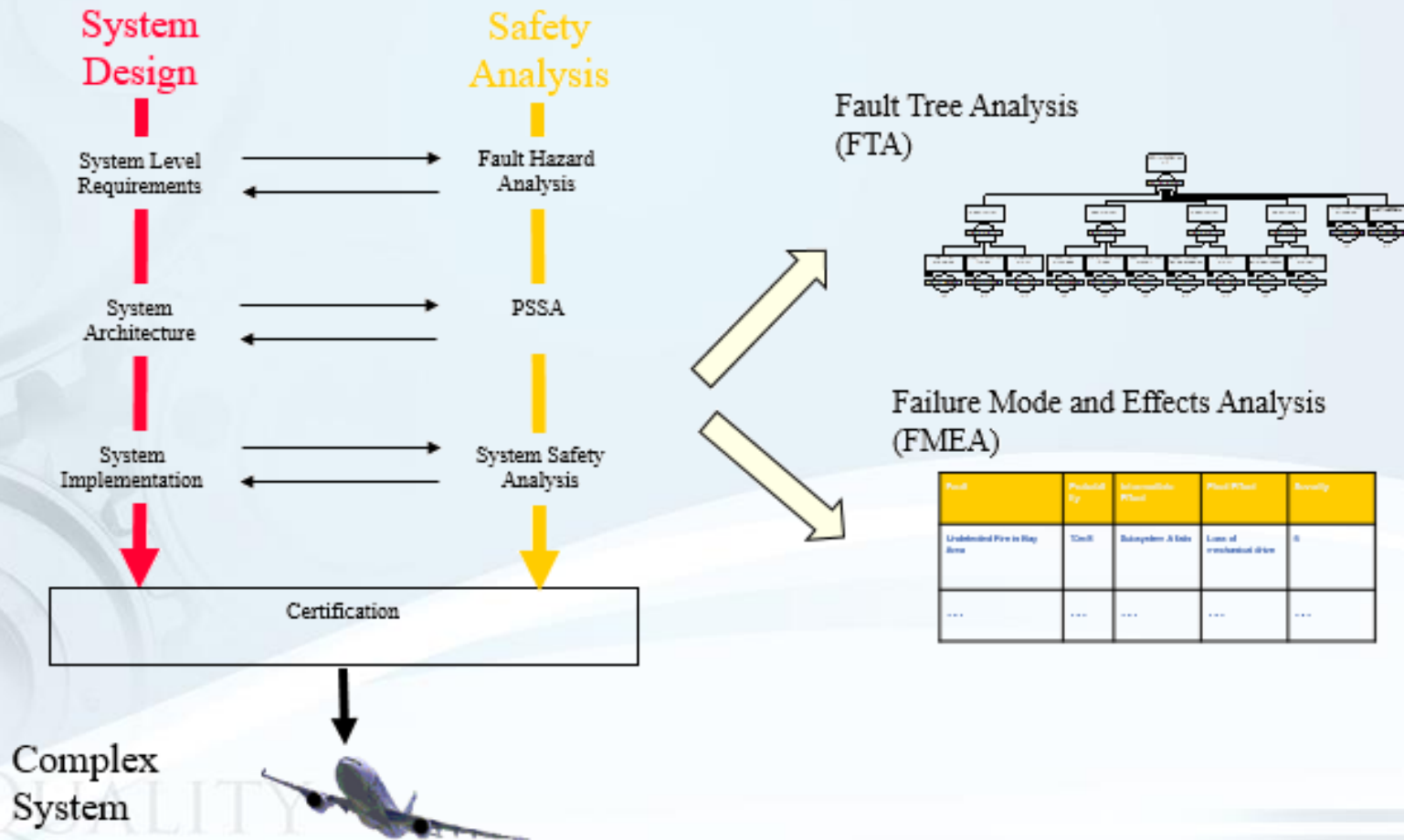


## Certain assumptions are to be considered with this methodology:

- Failure modes in FMEA are a direct result of the faults identified in the FTA process and the failure causes are assumed to be mutually independent.
- In the FMEA method, only the most critical failure modes are considered. Double or multiple failure modes inclusion, though representing a major improvement to traditional FMEA, would be important only when the assessment's aim is beyond the scope of this work such as risk identification and further quantitative analysis. O\*M\*N\*E\*X\*
- The complex system under consideration should be coherent and modular with each module relevant to system functioning, with the FTA possessing only AND and OR gates.

QUALITY

# Fault Tree Analysis VS FMEA



# Fault Tree Analysis VS FMEA

## FTA

- FTA is the “Top-Down” technique that is concerned with the identification and analysis of conditions that lead to the occurrence of a defined effect in contrast with the FMEA
- It is a EFFECT => CAUSE model

## FMEA

- FMEA is a “Bottom-up” technique which examines the failure mode of the components within the system and traces towards the potential effects of each component failure mode on system performance
- It is a CAUSE => EFFECT model

QUALITY

# Fault Tree Analysis VS FMEA

## FTA

- Consider using FTA rather than FMEA when you are particularly concerned about one or just a few system conditions that pose a unacceptable consequences
- FTA is very good at showing how robust a system will be to one or more initiating faults and for systems with high levels of redundancy /diversity for those with majority voting logic

## FMEA

- FMEA will be more appropriate than FTA when you suspect that large number of distinct system conditions with a range of unacceptable consequences
- FMEA is more suited to analysing systems that contain little or no redundancy and does not examine the effects of multiple failures at system level

QUALITY

# Fault Tree Analysis VS FMEA

## FTA

- FTA will identify combinations of conditions and component failures which will lead to single defined adverse effect

## FMEA

- FMEA on the other hand considers all single component failures in turn and identifies the range of their effects of the system

QUALITY

# Breakout Exercise 2

Create a the FMEA by Using FTA

QUALITY



# STEP 4

Evaluate the Fault Tree

QUALITY



# Evaluate Fault Tree

- Qualitative Analysis
  - Generate cut sets
  - Verify correctness of cut sets
  - Evaluate cut sets for design impact
- Quantitative Analysis
  - Apply failure data to tree events
  - Compute tree probability
  - Compute importance measures
  - Evaluate probability for design impact

Generate FT results and interpret the findings

QUALITY

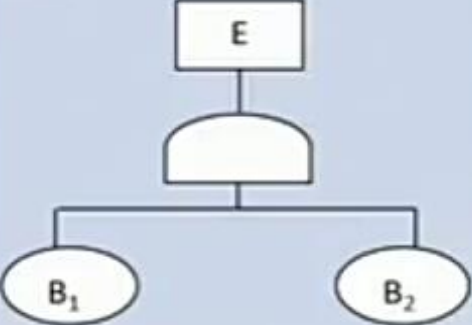
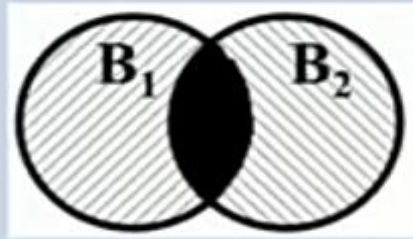
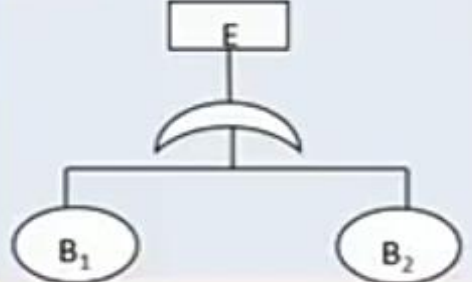
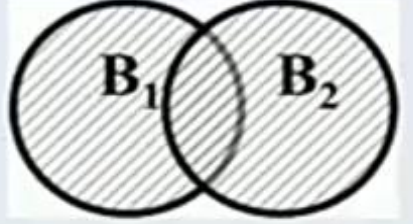


# Fault Tree Quantification

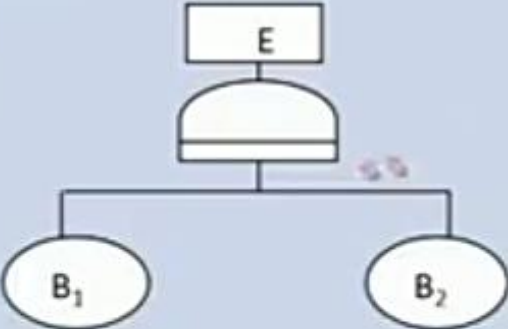
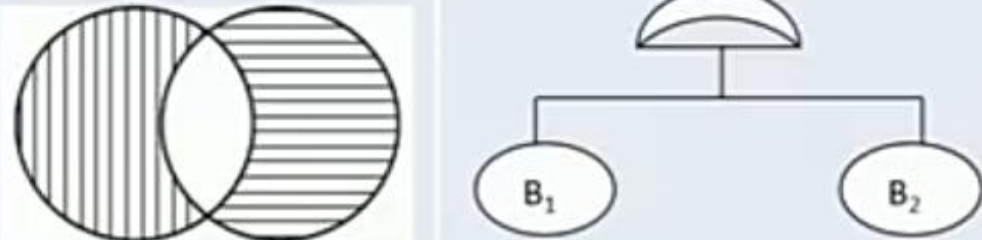
- The aim of fault tree quantification is to find out the probability of the top event to occur when the probability of the basic events occurrence are known.
- The basic events may be independent or dependent. The assumptions of independency make the mathematics simpler. Dependent basic events are the result of common cause failures.
- The two mostly used methods of quantification are –
  - 1) Gate-by-Gate Method.
  - 2) Cut sets Method.

QUALITY

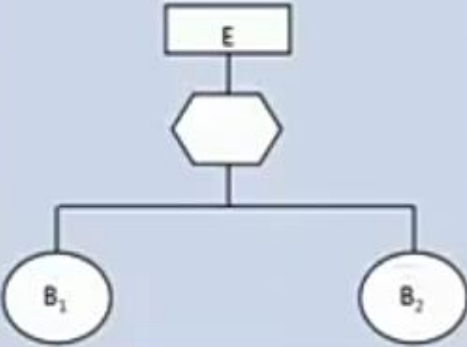
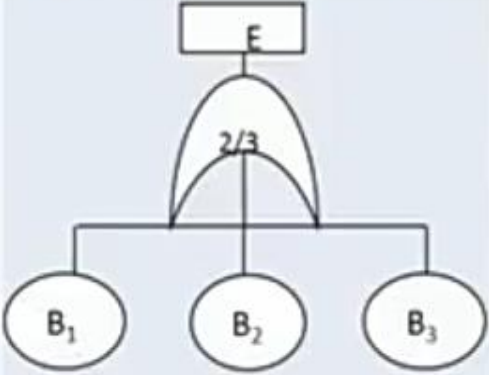
# Gate by Gate Method

Gate	Venn Diagram	Top event probability
AND 		$P(E) = P(B_1) \cdot P(B_2)$
OR 		$P(E) = P(B_1) + P(B_2) - P(B_1) \cdot P(B_2)$

# Gate by Gate Method

Gate	Representation	Top event probability
Priority AND		$P(E) = P(B_1) \cdot P(B_2)/2!$
Executive OR		$P(E) = P(B_1) + P(B_2) - 2P(B_1) \cdot P(B_2)$

# Gate by Gate Method

Gate	Gate representation	Top event probability
Inhibit Gate		$P(E) = P(B_1) \cdot P(B_2)$
Voting Gate		$P(E) = P(B_1) \cdot P(B_2) + P(B_2) \cdot P(B_3) + P(B_3) \cdot P(B_1) - 2P(B_1) \cdot P(B_2) \cdot P(B_3)$

# Breakout Exercise 3 (a)

## Probabilistic Risk Assessment Gate by Gate Method

QUALITY



# Probability of basic events failure

- Primary tank failure =  $10^{-3}$
- Primary contact failure =  $2 \times 10^{-3}$
- Primary timer failure =  $4 \times 10^{-3}$
- Primary switch failure =  $2 \times 10^{-4}$
- Primary operator failure =  $3 \times 10^{-4}$
- Primary alarm failure =  $3 \times 10^{-3}$
- Automatic valve malfunctioning =  $10^{-3}$

QUALITY



# Cut Set Method

- Gate by Gate method is applicable to small fault tree, we require to use computer programme using an efficient algorithm. Cut set method is used for this purpose.
- A set containing  $\{B_1, B_2, \dots, B_n\}$ , the collection of the all basic events of a fault tree, is termed as basic event.
- For the top event to occur it may not require all the events in the basic set to occur.
- A Cut set is a sub set of the basic set such that if all the basic events in the cut set occur, the top event will occur. So, the basic set is definitely a cut set.

QUALITY

# Identify the cut sets

- Risk is estimated for each event
  - When available, the failure rate data can be used to calculate the risk of a single chain or the many chains.
  - If there is no data, an estimate is established based on subjective guidelines similar to those used in FMEA development
- The Cut Sets with risk greater than the system can tolerate (i.e. safety or inoperative conditions) are selected for mitigation.
- Actions are required for Critical (red) and High Risks (orange)

QUALITY



# Cut set terms

- Cut Set
  - A set of events that together cause the tree Top UE event to occur
- Min CS (MCS)
  - A CS with the minimum number of events that can still cause the top event
- Super Set
  - A CS that contains a MCS plus additional events to cause the top UE
- Critical Path
  - The highest probability CS that drives the top UE probability
- Cut Set Order
  - The number of elements in a cut set
- Cut Set Truncation
  - Removing cut sets from consideration during the FT evaluation process
  - CS's are truncated when they exceed a specified order and/or probability

# Cut set

- A unique set of events that together cause the Top UE event to occur
- One unique root cause of the Top UE (of possibly many)
- A CS can consist of one event or multiple simultaneous events or elements

**Note:**

A CS element can be a:

- Failure
- Human error
- Software anomaly
- Environment condition
- Normal action

QUALITY

# The value of cut set

- CSs identify which unique event combinations can cause the UE
- CSs provide the mechanism for probability calculations
- CSs reveal the critical and weak links in a system design
  - High probability
  - Bypass of intended safety or redundancy features

**Note:**  
Always check all CS's against the system design to make sure they are valid and correct.

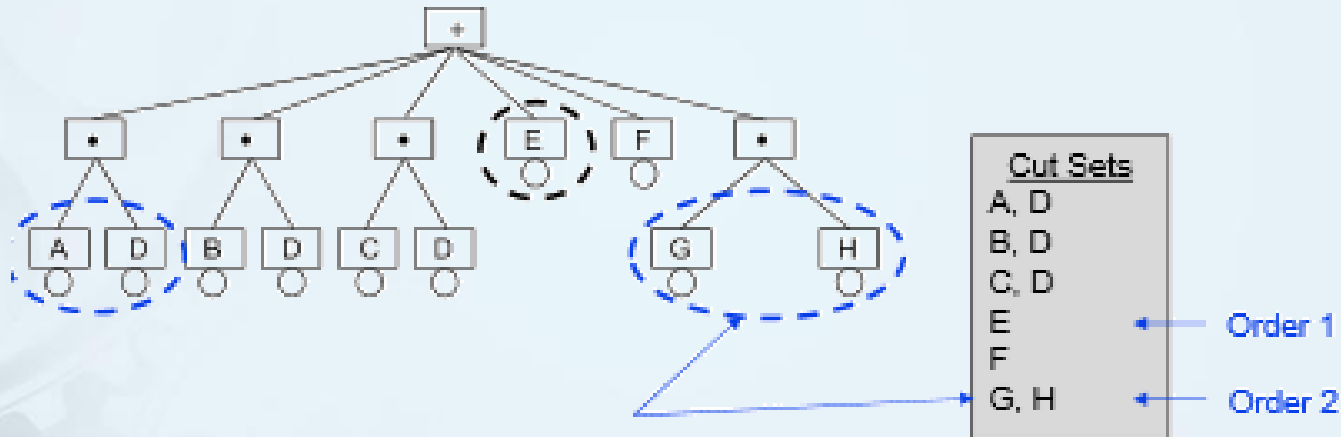
QUALITY

# MOCUS Algorithm

- MOCUS uses two principles.
  - Principle 1: An 'AND' gate increases the number of basic events in a cut set.
  - Principle 2: An 'OR' gate increases the number of cut set.
- The step by step procedure of MOCUS algorithm is given below.
  - Step 1 : Alphabetized each gate and number each basic events.
  - Step 2 : Consider the upper most gate first. Identify all the input to this gate.

QUALITY

# Cut sets

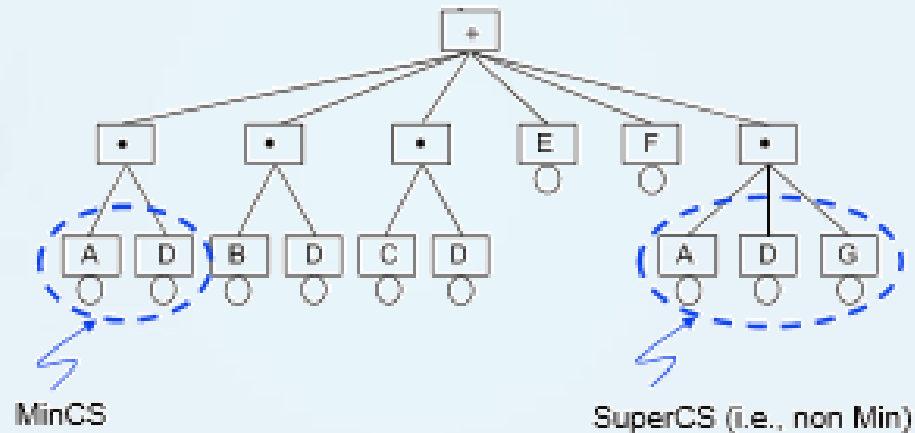


AND gate means that both G & H must occur. Since they go directly to top, they comprise a CS, denoted by {G, H}.

## Cut Set (CS)

A unique set of events that cause the Top UE to occur.

# Min CS



## Min CS

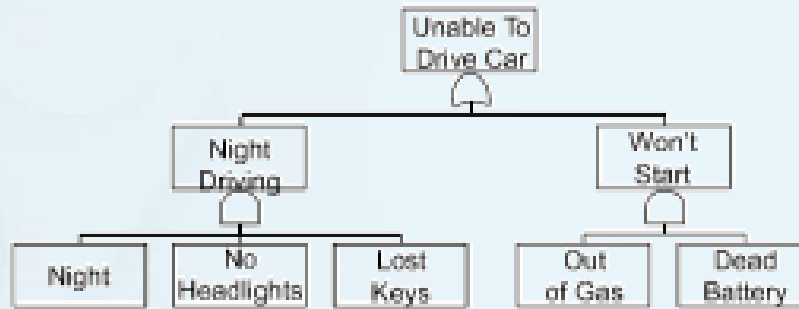
A set of events that contain the minimum number of *necessary* events to cause the Top UE; it cannot be further reduced.

## Super CS

A set of events that contain a number of events *sufficient* to cause the Top UE (ie, more than necessary as a minimum).

QUALITY

# Min CS - Example



If an item can be removed from CS and top still occurs then its not a Min CS.

CS1 - Night & No Headlights & Lost Keys

CS2 - Out of Gas & Dead Battery

Invalid FT  
(Not Min CS's)

Should be:

Night & No Headlights

Lost Keys

Out of Gas

Dead Battery

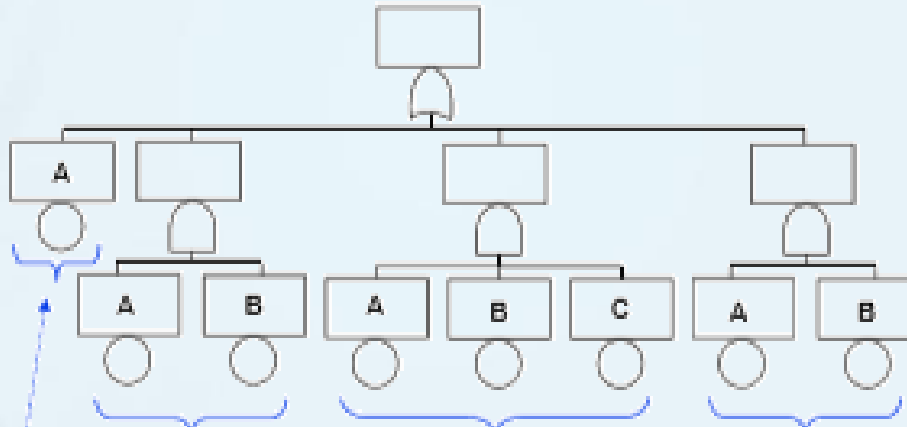
# Min CS

- A CS with the minimum number of events that can still cause the top event
- The true list of CS's contributing to the Top
- The final CS list after removing all SCS and DupCS
- Additional CS's are often generated, beyond the MinCS's
  - Super Cut Sets (SCS) – result from MOE's
  - Duplicate Cut Sets (DupCS) - result from MOE's or AND/OR combinations
- Why eliminate SCS and DupCS?
  - Laws of Boolean algebra
  - Would make the overall tree probability slightly larger (erroneous but conservative)

QUALITY



# Min CS



Cut Sets:

A

A,B

A,B,C

A,B

← SCS

← SCS

← DupCS, SCS



Min Cut Sets:

A

QUALITY

# Breakout Exercise 3(b)

## Probabilistic Risk Assessment Cut Set Method

QUALITY



# STEP 5

Control the Undesired Event (Hazard)

QUALITY

# Mitigate the risk

Risk Mitigation can take many forms. A popular method is to use the criticality method. Other techniques require a level of mitigation calculated to Defects per Million Opportunities (DPMO).

Safety systems may require resulting risk to be mitigated to:  
Error Proofing (cannot Occur)  
1 in 10 million (1 X 10 to the minus 7)

Action logs and revision records are kept for follow-up and closure of each undesirable risk.

QUALITY

# Mitigate the risk

Any risk not mitigated to an acceptable level is a candidate for Mistake Proofing or Quality Control, which protects the consumer from the risk

QUALITY



# Examples of mitigation strategies

When a risk is unacceptable the team may have several options available. The following are a few examples of the options available:

## Design change

Selection of a component with a higher reliability to replace the

## Base-level event component

This is often expensive unless identified early in Product Development

QUALITY

# Examples of mitigation strategies

## **Physical Redundancy of the Component**

This option places the redundant component in parallel to the other. Both must fail simultaneously for the hazard to be experienced. If a safety issue exists, this option may require non-identical components

## **Software Redundancy**

The addition of a sensing circuit, which can change the state of the product, often reduces the severity of the event by protecting components through duty cycle changes and reducing input stresses when identified.

QUALITY

# Examples of mitigation strategies

## **Warning System**

The circuit may just warn of an event. This requires action by an operator or analyst. It is important to note that if this course of action is taken, Human Factors Reliability must also enter the evaluation.

## **Quality Control**

This may include removal of the potential failure through testing or inspection. The inspection effectiveness must match the level of severity that the hazard may impose on the consumer.

QUALITY

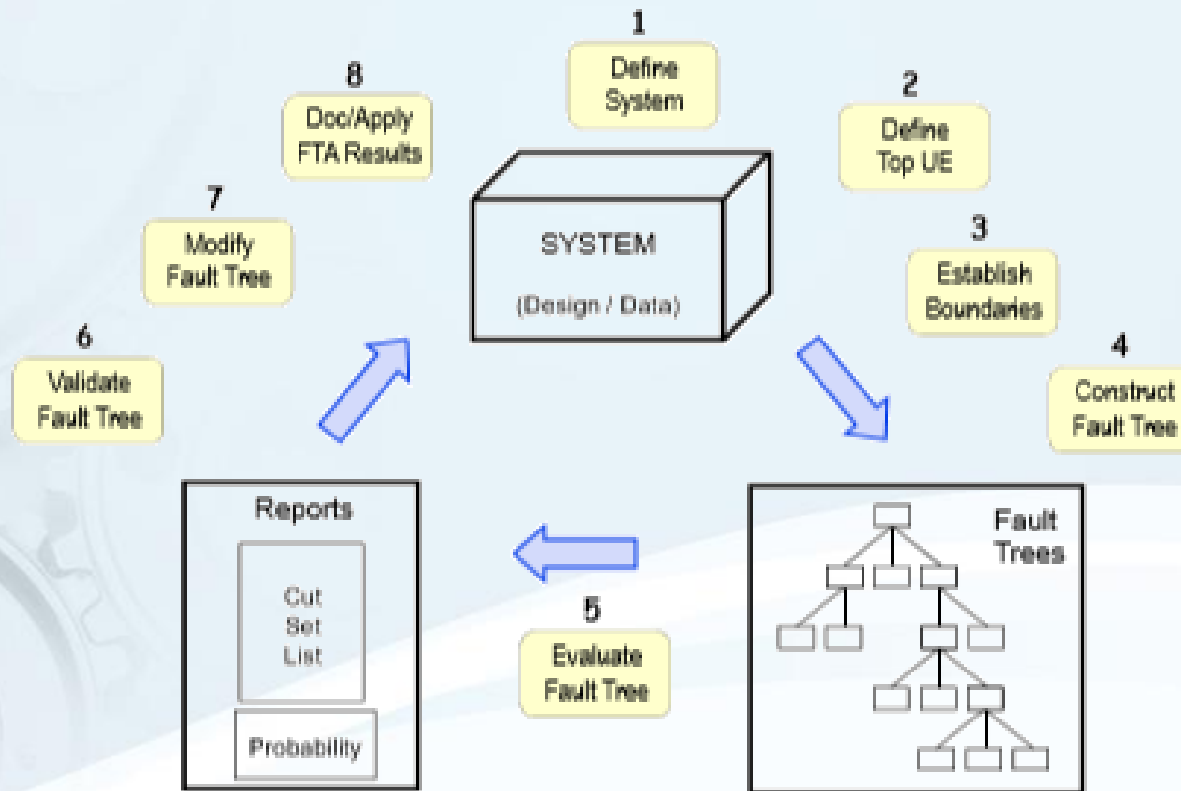


# FTA SUMMARY

QUALITY



# FTA Summary



QUALITY

# FTA Summary

- FTA is an **analysis tool**

- Strengths – methodical, structured, graphical, quantitative, easy to model complex systems
- Coverage – hardware, software, humans, procedures, timing
- Like any tool, the user must know when, why and how to use it correctly

- FTA is for **system evaluation**

- Safety – hazardous and catastrophic events
- Reliability – system unavailability
- Performance – unintended functions

- FTA is for **decision making**

- Root cause analysis
- Risk assessment
- Design assessment

*Thank You!*

*Questions?*



**info@omnex.com**  
**734.761.4940**

