

Root Cause Analysis

For Anand Group

QUALITY

© 2017, Omnex, Inc.
315 Eisenhower Parkway Suite 214
Ann Arbor, Michigan 48108
USA
734-761-4940
Fax: 734-761-4966

First Edition
JUNE 2019

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.



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

Web: www.omnex.com

QUALITY



Course Objectives

- Understand the concept and purpose of RCA
- Understand the purpose of FTA
- Understand the different symbols used in FTA
- Demonstrate an ability to construct and effectively complete the FTA
- Explain the relationship between the FTA and FMEA

QUALITY



Agenda

- Chapter 1 – Introduction of RCA
- Chapter 2 – Introduction of FTA
- Chapter 3 – Understanding symbols of FTA
- Chapter 4 – Development of FTA
 - **Breakout Exercise 1: Identify the hazard & Understanding the system.**
 - **Breakout Exercise 2: Create a fault tree.**
 - **Breakout Exercise 3: Probabilistic Risk Assessment.**
 - **Breakout Exercise 4: Development of Risk Mitigation.**

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).

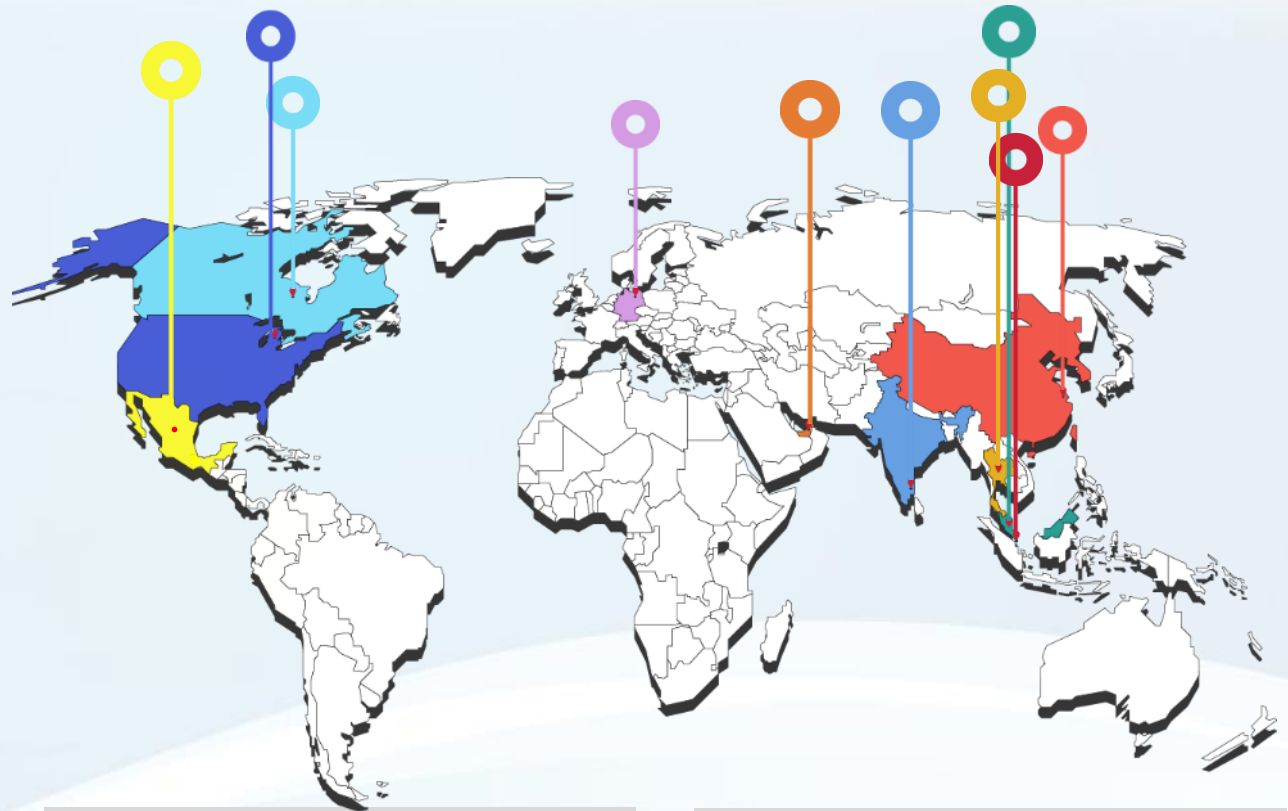




Omnex is headquartered and operates from the United States through offices in Michigan.

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
info@omnex.com



● Omnex Global Head Quarters (Michigan, USA)
West Coast Operations (San Jose, CA)

● Asia Pacific HQ (Chennai, Pune, Delhi, Bangalore)

● China (Shanghai, Guangzhou, Wuhan, Chengdu)

● Canada (Mississauga)

● Europe (Berlin, Germany)

● Middle East (Dubai, Saudi Arabia, Bahrain)

● Thailand (Bangkok)

● Mexico (Monterrey)

● Singapore

● 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
- ✓ 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 Root Cause Analysis using FTA?
 - What are your experiences with respect to FTA?
 - Please share something unique and/or interesting about yourself.



QUALITY

Chapter 1

Introduction of Root Cause Analysis

QUALITY

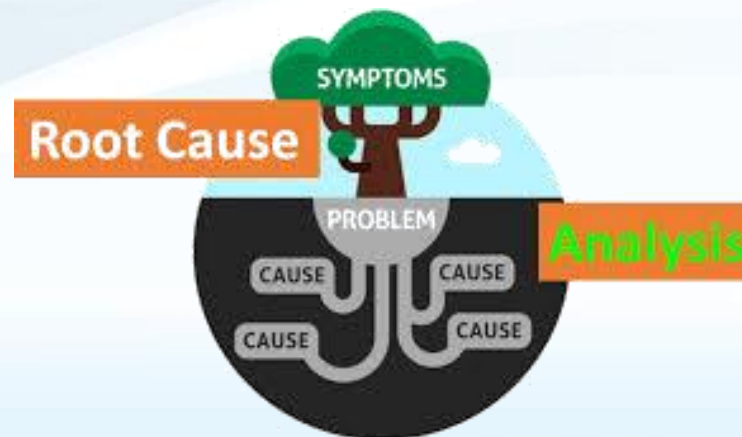
What is a Root Cause?

- Root cause is the underlying or deep or fundamental reason for the occurrence of the problem.
- Root cause is defined as a factor which caused the non conformance and should be eliminated through process improvement.
- Root cause is the deepest cause in the casual chain that leads to a problem or outcome.



What is Root Cause Analysis?

- Root Cause Analysis is a systematic process for defining , understanding , analysing , identifying root causes and an approach for solving the problem.
- Root Cause Analysis is defined as a collective term that describes a wide range of approaches, tools, to identify the real causes of non conformance or the problem.



Approaches to Root Cause Analysis

- **Events and causal factor analysis**

Widely used for major, single-event problems, such as a refinery explosion, this process uses evidence gathered quickly and methodically to establish a timeline for the activities leading up to the accident.

- **Change analysis**

This approach is applicable to situations where a system's performance has shifted significantly. It explores changes made in people, equipment, information, and more that may have contributed to the change in performance.

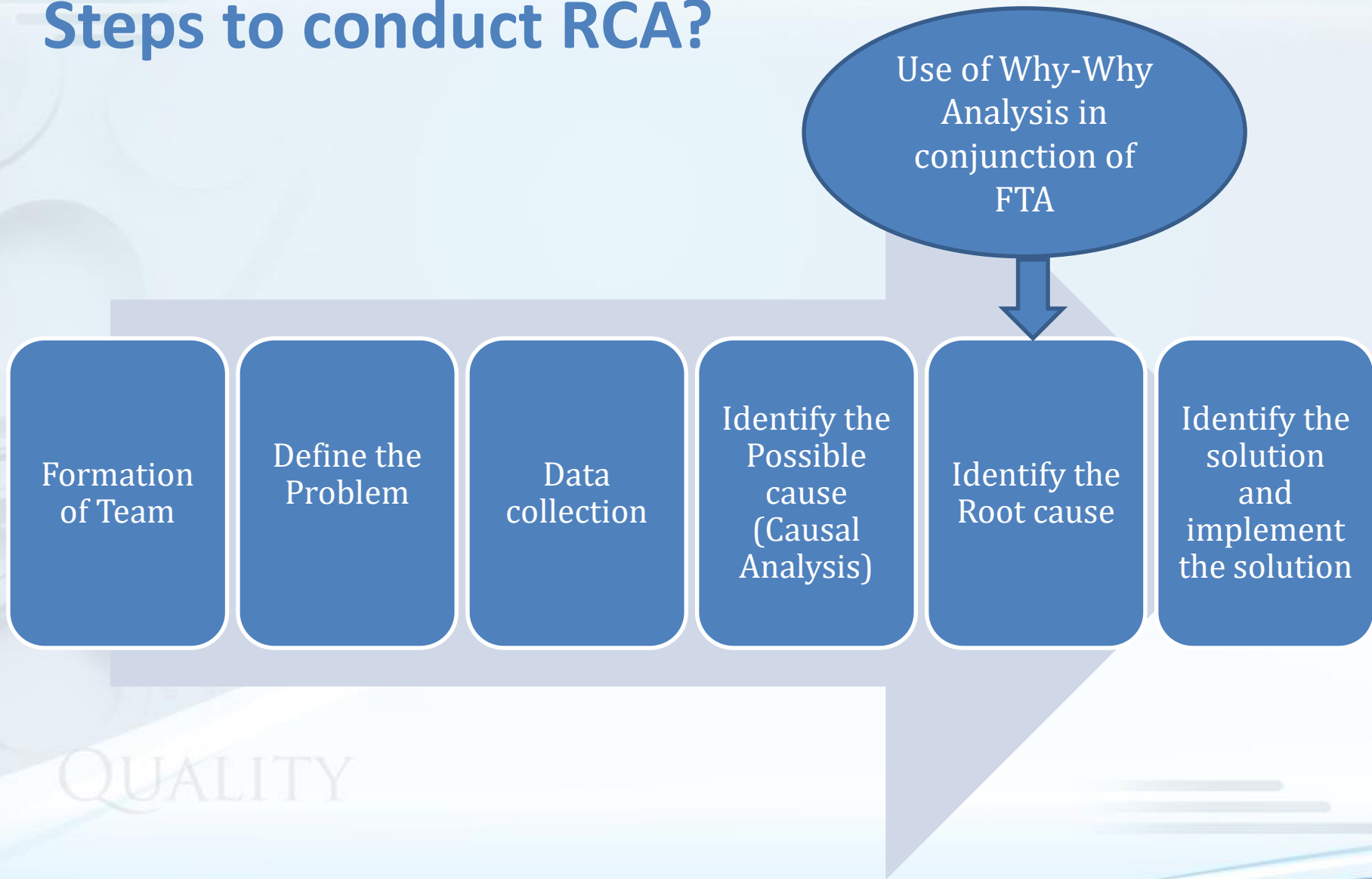
- **Barrier analysis**

This technique focuses on what controls are in place in the process to either prevent or detect a problem, and which might have failed.

- **Management oversight and risk tree analysis**

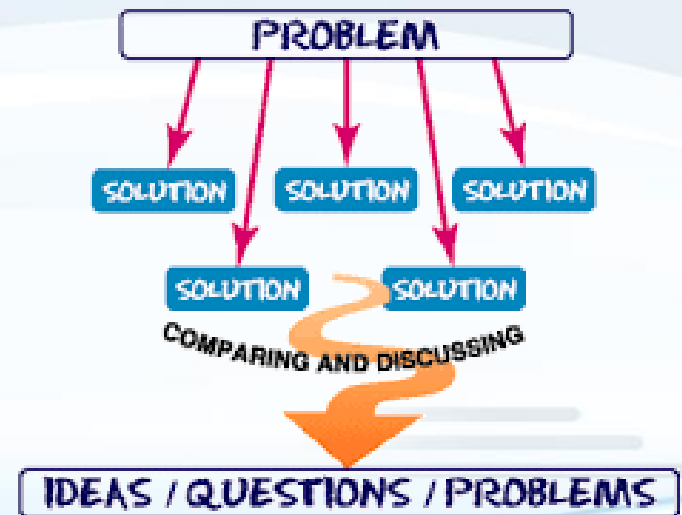
One aspect of this approach is the use of a tree diagram to look at what occurred and why it might have occurred.

Steps to conduct RCA?



What is Problem?

- Problem is a situation preventing something from being achieved
- Problem is derived from the Greek Word meaning “obstacle”
- Problem is the difference between the planned result and what is achieved actually (if the target is not achieved)
- Problem is an opportunity for improvement
- Problem is the gap between the current state and the goal state



Data Collection

- Data collection is the process of gathering and measuring information on variable of interest, in an established systematic fashion that enables one to answer stated questions.
- Regardless of the field of study or preference for defining data (quantitative, qualitative) accurate data collection is essential to maintaining the integrity of study.
- The primary rationale for preserving the data integrity is to support the detection of error in the data collection process.

QUALITY



Some of the tools for data collection

- Check sheet
- Histogram
- Pareto Diagram
- Run chart
- Scatter Diagram

QUALITY



Generate the Possible Cause

Examine Changes

- How could this change possibly cause the problem?
- Be creative, yet realistic.
- Could a change have multiple effects?

QUALITY

Identify the Possible Cause

- Perform a comparison analysis to determine if the same or similar problem existed in related products or processes.
 - Identify past solutions and root causes that may be appropriate for the current problem.
- Identify the top few potential causes; develop a plan for investigating each cause and update the action plan.
- Evaluate a potential cause against the problem description. Does a mechanism exist so that the potential cause could result in the problem?

Analyse the Possible Cause

- Use the iterative process to analyze each potential cause:
 - **Hypothesis Generation** – How does the potential cause result in the problem?
 - **Design** – What type of data can most easily prove/disprove the hypothesis?
 - **Preparation** – Obtain materials and prepare a checklist.
 - **Data Collection** – Collect the data.
 - **Analysis** – Use simple graphical methods to display data.
 - **Interpretation** – Is the hypothesis true?
- Investigate several potential causes independently.
- Use an action plan to manage the analysis process for each potential cause being studied.

Confirm the Cause

- Does this cause explain all that is known about what:
 - The problem is?
 - What the problem is not?

Test Cause Work sheet						
Concern Number: 2012-008						
Concern Title: Rusty Shafts						
POSSIBLE CAUSES						
IS A NOT CRITERIA	Coolant No Good	pH Too Low in Washer	Rust From Bar Stock	Packed Hot	Heat Treat	Separator
XJ5056 Shaft	-	+	+	+	+	+
Line of Rust	-	-	-	-	-	+
Surface Rust	+	+	-	+	+	+
Random	-	-	-	+	-	+
In Shipping	-	+	-	+	-	+
Boxes	-	+	-	+	-	+
7/28/2012	-	+	-	+	-	-
#= 17290	+	+	+	+	+	+
One Defect	+	+	+	+	+	+
A + indicates that a Is/A Not criterion produces an affirmative answer to the Key Question. A - indicates that a Is/A Not criterion produces a negative answer to the Key Question. A ? indicates that a Is/A Not criterion requires investigation.						
Key Question: Does this cause explain all that is known about what the problem is and all that is known about what the problem is not?						

QUALITY



Confirm the Cause

Confirm Root Causes

Test Cause Worksheet						
Concern Number: 2012-008						
Concern Title: Rusty Shafts						
POSSIBLE CAUSE S						
IS/IS NOT CRITERIA	Coolant No Good	pH Too Low in Washer	Rust From Bar Stock	Packed Hot	Heat Treat	Separator
XJ5056 Shaft	-	+	+	+	+	+
Line of Rust	-	-	-	-	-	+
Surface Rust	+	+	-	+	+	+
Random	-	-	-	+	-	+
In Shipping	-	+	-	+	-	+
Boxes	-	+	-	+	-	+
7/28/2012	-	+	-	+	-	-
#=17290	+	+	+	+	+	+
One Defect	+	+	+	+	+	+
<p>A + indicates that a Is/Is Not criterion produces an affirmative answer to the Key Question. A - indicates that a Is/Is Not criterion produces a negative answer to the Key Question. A ? Indicates that a Is/Is Not criterion requires investigation.</p> <p>Key Question: Does this cause explain all that is known about what the problem is and all that is known about what the problem is not?</p>						

QUALITY



Root Cause Analysis

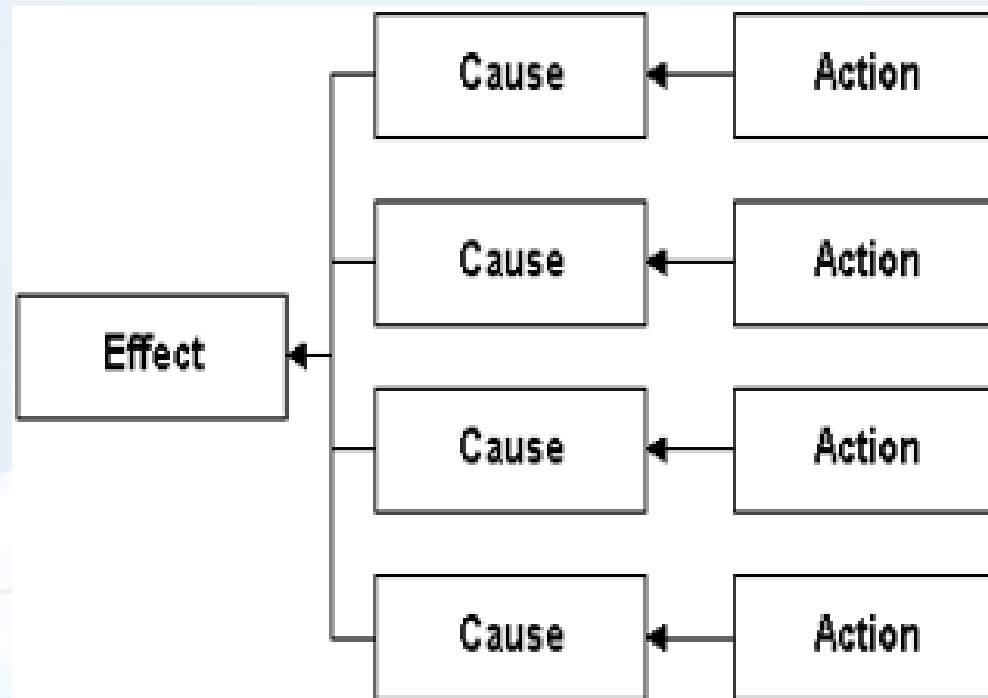
- Conduct Root cause analysis by using Why-Why Analysis.
- FTA depicts the risk-based path to a root cause or Base-level event.
- When investigating a failure, the chain of events depicted by FTA allows the problem solver to see the events leading to a root cause(s) or Base-level event.
- Mainly used in Investigation of a safety or regulatory concern.
- Used in Reliability Engineering.

QUALITY

Identify the Solution

Brainstorm Potential Corrective Actions

- For each potential root cause



Implement the Solution

1. Elimination of Root Cause
2. Establish Givens/Wants for Corrective Action (Poka-Yoke)
3. Verified Root Cause
4. Brainstorm Alternate Corrective Actions
5. Select Best Choice
6. Consider Risks Involved In Selected Action
7. Verify Corrective Action: (Before & After)

QUALITY

Verify the Solution

Verification Questions

- Has the customer been contacted to determine a date when verification will be evaluated?
- What data has been established for follow-up?
- Has a timeline (project) chart been completed?
- Have field tests involved customer groups?
- Have dates been established when verification of effectiveness will be evaluated?

Not acceptable to depend on the customer for verification activities

What is Problem Solving?

- Problem solving is the act of defining a problem; determining the cause of the problem, identifying, prioritizing and selecting alternatives for a solution and implementing a solution.
- Problem solving is a process of working through details of a problem to reach a solution.
- Problem solving may include mathematical or systematic operations and can be a gauge of an individual critical thinking skills.

QUALITY

Where it is used?

Root Cause Analysis is used in the problem solving process to find out the root causes of the problem, identifying, prioritizing and selecting alternatives for a solution and implementing the solution.

QUALITY

Scope of problem solving?

- New Product Development.
- Serial Production.
- Post Production.
- Any Undesired events.

QUALITY



Chapter 2

Introduction of Fault Tree Analysis

QUALITY

Introduction of Fault Tree Analysis

- Fault tree Analysis was originally developed in 1962 at Bell laboratories by H.A. Watson
- Fault tree Analysis is one of the most important logic and probabilistic techniques used in Probabilistic Risk Assessment (PRA) and system reliability assessment
- FTA attempts to model and analyse failure processes of engineering systems .It can be simply described as a analytical technique
- FTA is a top down deductive analysis approach for resolving an undesired event into its causes

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

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
- **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

QUALITY

Stages of Problem solving

- **Proactive FTA**

- FTA during system design development
- Improve design by mitigating weak links in the design
- Prevent undesired events and mishaps

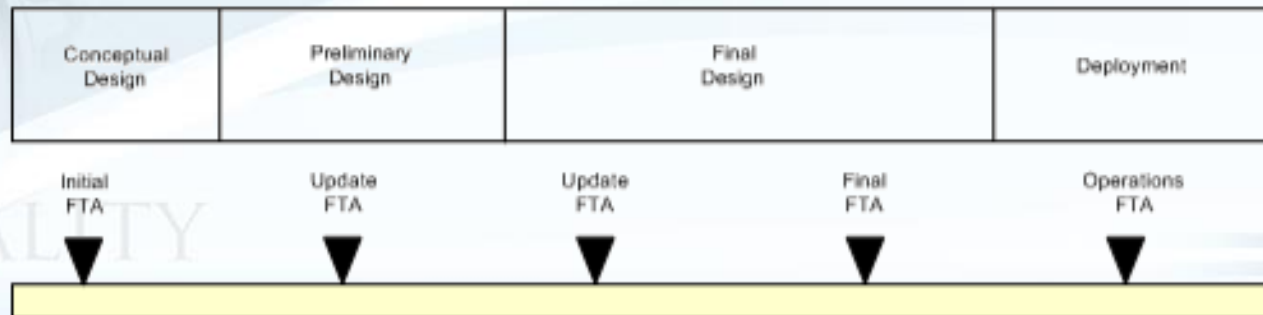
- **Reactive FTA**

- FTA during system operation
- Find root causes of a mishap/accident
 - Modify the design to prevent future similar accidents

QUALITY

What is the timeline of FTA

- Design Phase
 - FTA should start early in the program
 - The goal is to influence design early, before changes are too costly
 - Update the analysis as the design progresses
 - Each FT update adds more detail to match design detail
 - Even an early, high level FT provides useful information
- Operations Phase
 - FTA during operations for root cause analysis
 - Find and solve problems (anomalies) in real time



Example for FTA Application

- Evaluate inadvertent arming and release of a weapon
- Calculate the probability of a nuclear power plant accident
- Evaluate an industrial robot going astray
- Calculate the probability of a nuclear power plant safety device being unavailable when needed
- Evaluate inadvertent deployment of jet engine thrust reverser
- 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

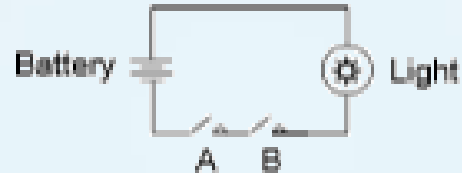
Starting point of FTA?

- Fault Tree is the logic model of the relationship of the undesired event to more basic events
- The top event of the fault tree is the undesired event
- The middle events are the intermediate events and the basic events are at the bottom
- The logic relationship of events are shown by logic symbols or gates

QUALITY

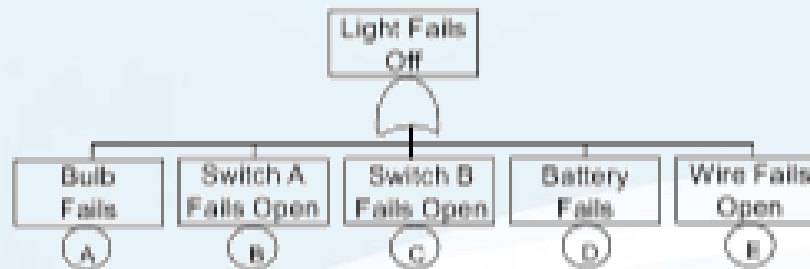
Example of FTA

System



System Undesired Event: Light Fails Off

FT Model

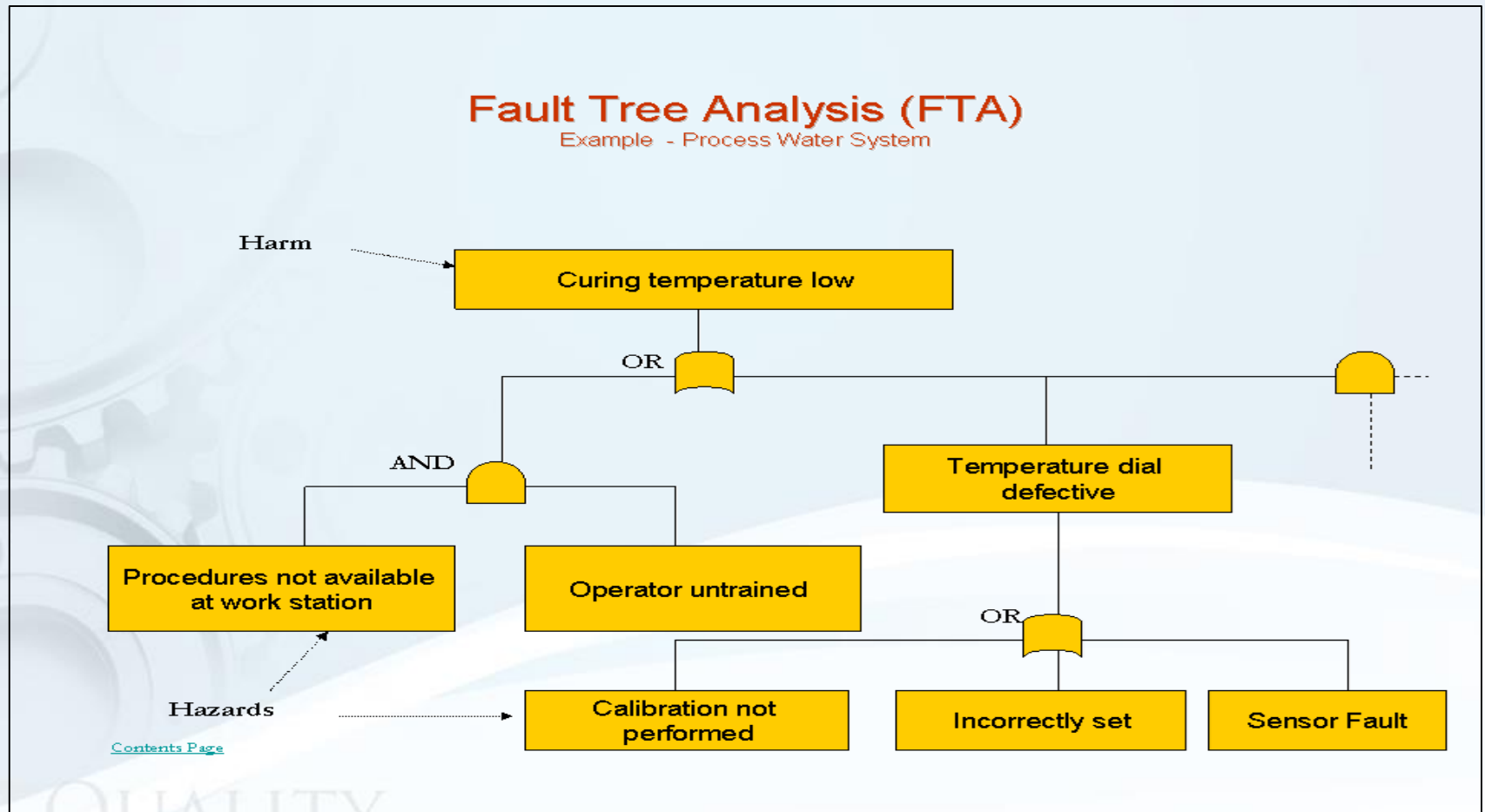


Cut Sets

Event combinations that can cause Top Undesired Event to occur

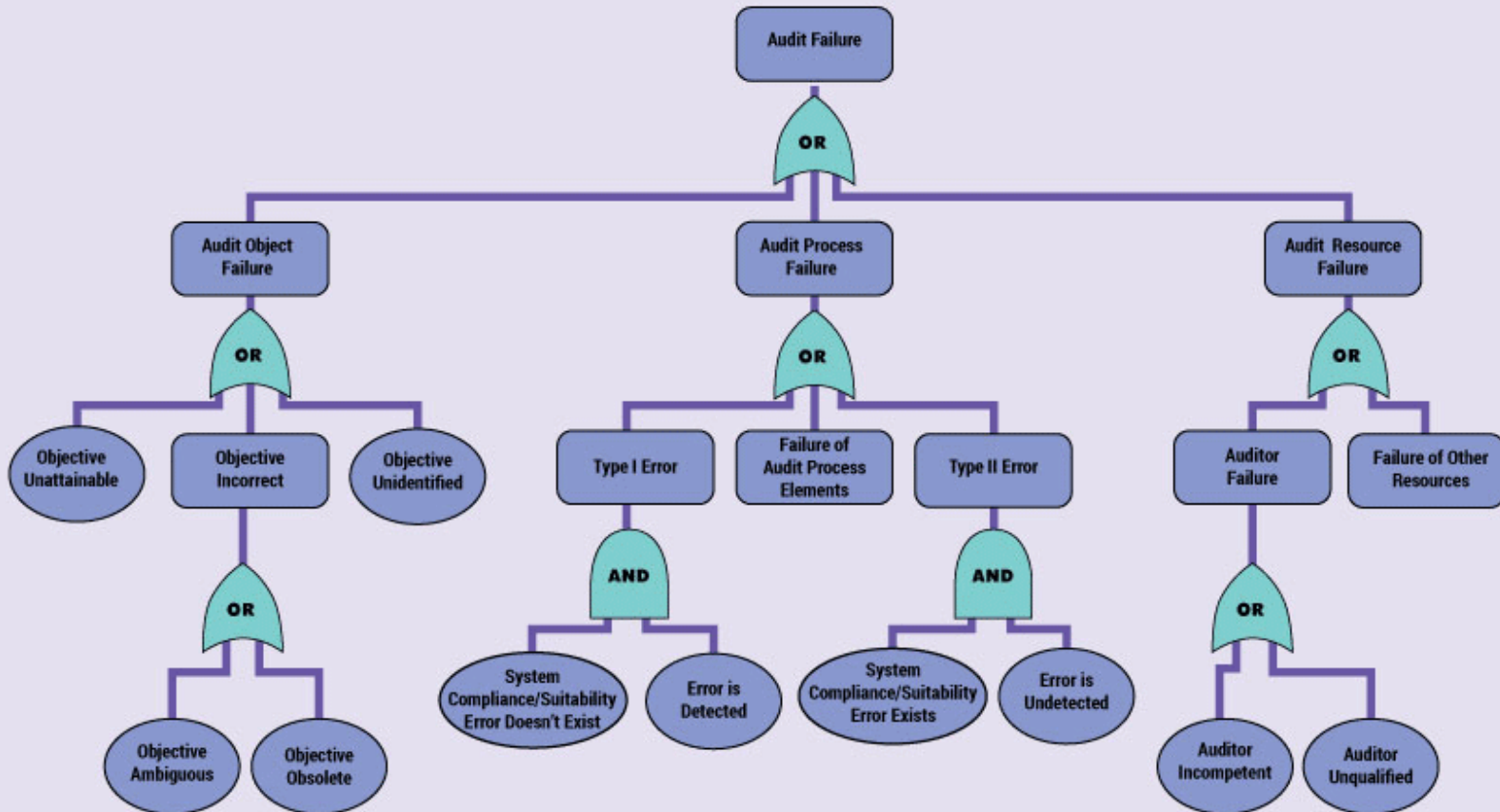
CS	Probability
A	$P_A=1.0 \times 10^{-6}$
B	$P_B=1.0 \times 10^{-7}$
C	$P_C=1.0 \times 10^{-7}$
D	$P_D=1.0 \times 10^{-6}$
E	$P_E=1.0 \times 10^{-6}$

Fault Tree Analysis Format – Example 1



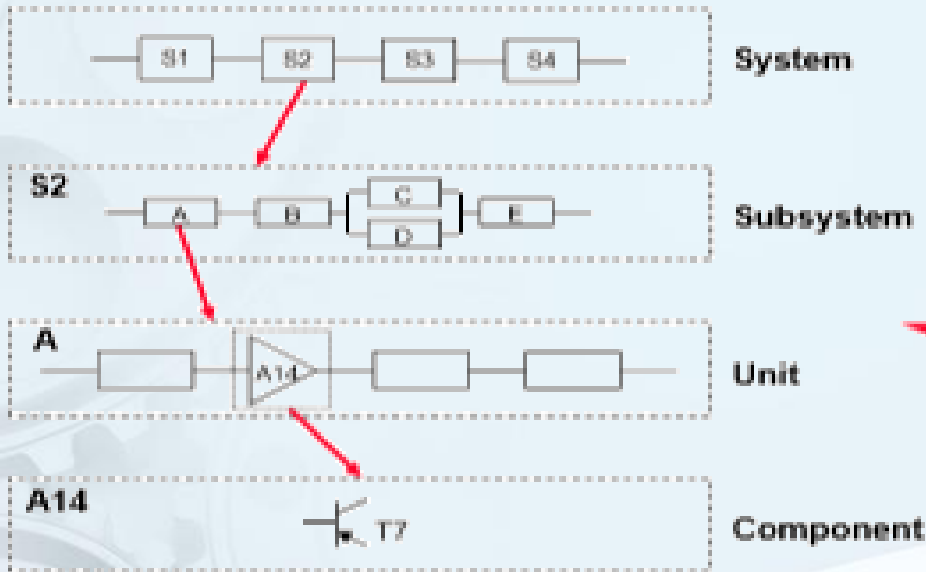
Fault Tree Analysis – Audit Failure Example 3

Fault Tree Diagram Example



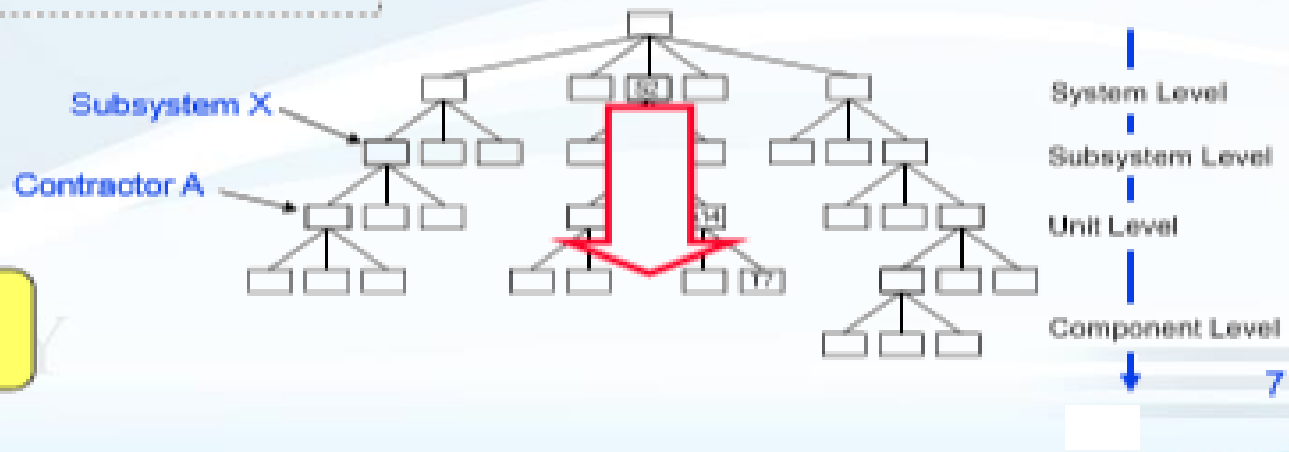
©2017 Creative Safety Supply

FTA Deductive approach



Going from the general to the specific.

Analyzing from the Undesired Event to the root cause(s).



Only the components that contribute to UE.

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
 - Interface analysis - contractors, subsystems
- 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

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
 - ☞ Carefully develop a name scheme - events, MOE's, transfers
- **Large FT's require more design foresight**
 - Develop organized plan when several analysts work on same FT

Chapter 3

Understand the symbol of FTA

QUALITY

FTA Building Block



Primary Failure



Secondary Failure



Normal Event

Basic Events



OR Gate



AND Gate



Inhibit Gate



Exclusive OR Gate



Priority AND Gate

Gates



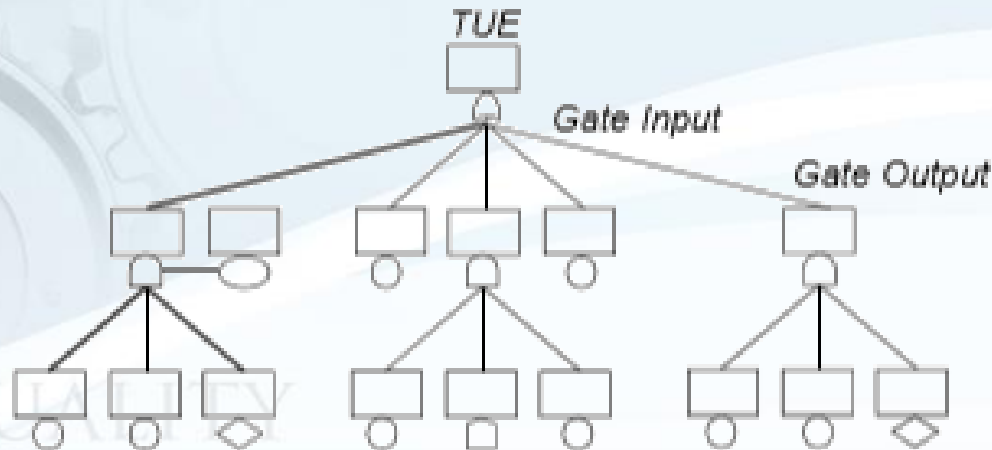
Text Box



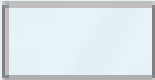
Condition






Transfer



Basic Fault Tree symbol

Symbol	Action	Description
	Text Box	Contains the text for a tree node

Tree Node

Symbol	Action	Description
	Primary Failure	Basic primary component failure mode
	Secondary Failure	a) Secondary component failure mode b) Event that could be further expanded
	Normal Event	An event that is normally expected to occur

Basic Events

Basic Events (BEs)

- Failure Event
 - Primary Failure - basic component failure (circle)
 - Secondary Failure - failure caused by external force (diamond)
- Normal Event
 - An event that describes a normally expected system state
 - An operation or function that occurs as intended or designed, such as "Power Applied At Time T1"
 - The Normal event is usually either On or Off, having a probability of either 1 or 0
 - House symbol

The BE's are where the failure rates and probabilities enter the FT

Event symbol Example

Resistor R77
fails open



Circle
Primary Failure

Basic inherent component failure

Resistor R77 fails
open from excessive
RF energy



Diamond
Secondary Failure

A) Failure caused by external force

Computer CC107
fails to operate



Diamond
High Level Failure

B) Failure that could be further developed

System power is
applied at T=100



House
Normal Event

An event that would occur under normal
Operation (without failure)






QUALITY

Gate Events (GEs)

- A logic operator combining input nodes
- Five basic logic operator types
 - AND, OR, Inhibit, Priority AND and Exclusive OR
 - Additional types do exist, but usually not necessary
- Represents a fault state that can be further expanded


QUALITY

Gate Symbol

Symbol	Action	Description
	OR Gate	The output occurs only if at least one of the inputs occur
	AND Gate	The output occurs only if all of the inputs occur together
	Inhibit Gate	The output occurs only if the input event occurs and the attached condition is satisfied
	Exclusive OR Gate	The output occurs only if at least one of the inputs occurs, but not both
	Priority AND Gate	The output occurs only if all of the inputs occur together, but in a specified sequence (input 1 must occur before 2)

Condition Events (CEs)

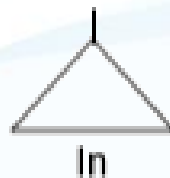
- A condition attached to a gate event
- It establishes a condition that is required to be satisfied in order for the gate event to occur

Symbol	Action	Description
	Condition Event	A conditional restriction or an event probability

QUALITY

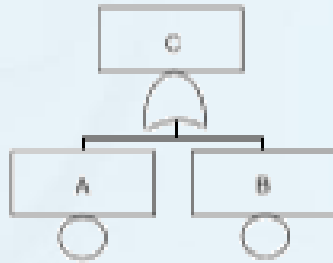
Transfer Event (TE)

- Indicates a specific tree branch (subtree)
- A pointer to a tree branch
- A Transfer only occurs at the Gate Event level
- Represented by a Triangle
- The Transfer is for several different purposes:
 - Starts a new page (for FT prints)
 - It indicates where a branch is used numerous places in the same tree, but is not repeatedly drawn (Internal Transfer)
 - It indicates an input module from a separate analysis (External Transfer)



QUALITY

OR GATE



Fault Tree

A	B	C
0	0	0
1	0	1
0	1	1
1	1	1

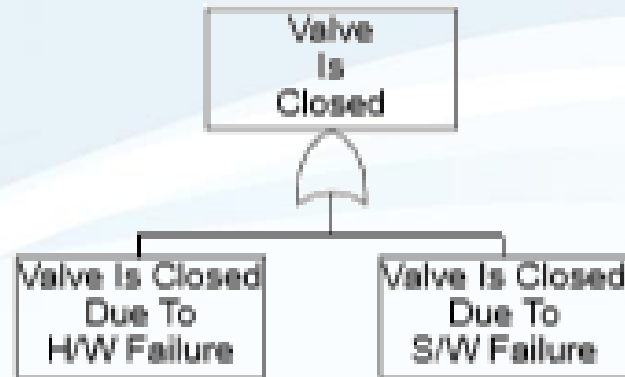
Truth Table

- Either A or B is necessary and sufficient to cause C
- Both A and B can occur together to cause C
- Example: Light is off because light bulb fails OR power fails

QUALITY

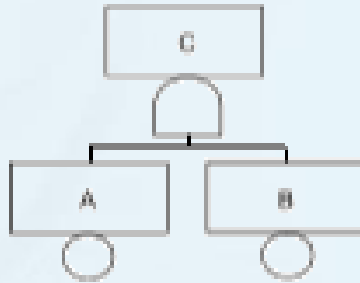
OR GATE

- Causality passes through an OR gate
 - Inputs are identical to the output, only more specifically defined (refined) as to cause
 - The input faults are never the cause of the output fault
 - ◆ Passes the cause through
 - ◆ Not a cause-effect relationship



QUALITY

AND GATE



Fault Tree

A	B	C
0	0	0
1	0	0
0	1	0
1	1	1

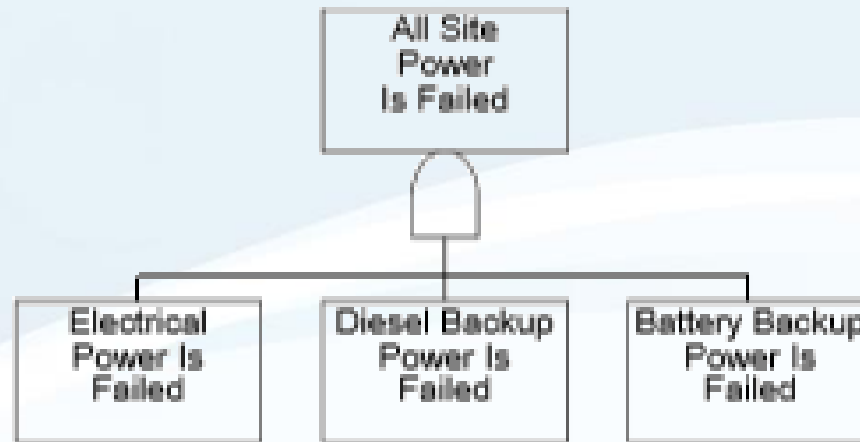
Truth Table

- Both A and B are necessary to cause C
- A and B must occur simultaneously
- Example: No power available because Primary power fails AND Secondary power fails

QUALITY

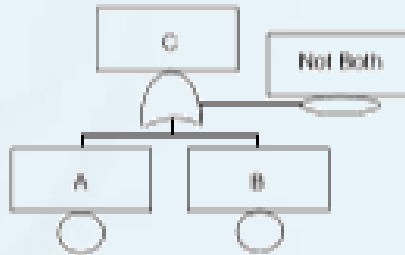
AND GATE

- Specifies a causal relationship between the inputs and the output
 - Causality is created at the AND gate
 - The input faults collectively represent the cause of the output fault
 - Implies nothing about the antecedents of the input faults



QUALITY

Exclusive OR GATE



Fault Tree

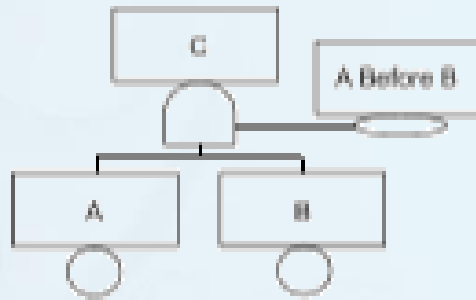
A	B	C
0	0	0
1	0	1
0	1	1
1	1	0

Truth Table

- Either A or B is necessary and sufficient to cause C
- But, both A and B cannot occur together (at same time)
- Only allow two inputs (cascade down for more ExOR inputs)
- Example: Relay is energized OR Relay is de-energized, but not both

QUALITY

Priority AND GATE



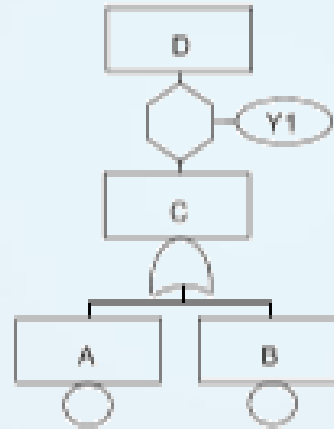
Fault Tree

A	B	C
0	0	0
1	0	0
0	1	0
1	1	1

Truth Table

- Both A and B are necessary to cause C
- But, A must occur before B
- Show priority order with inputs from left to right
- Example: Fault is not detect because Monitor fails before Computer fails




Inhibit GATE



Effectively an AND gate

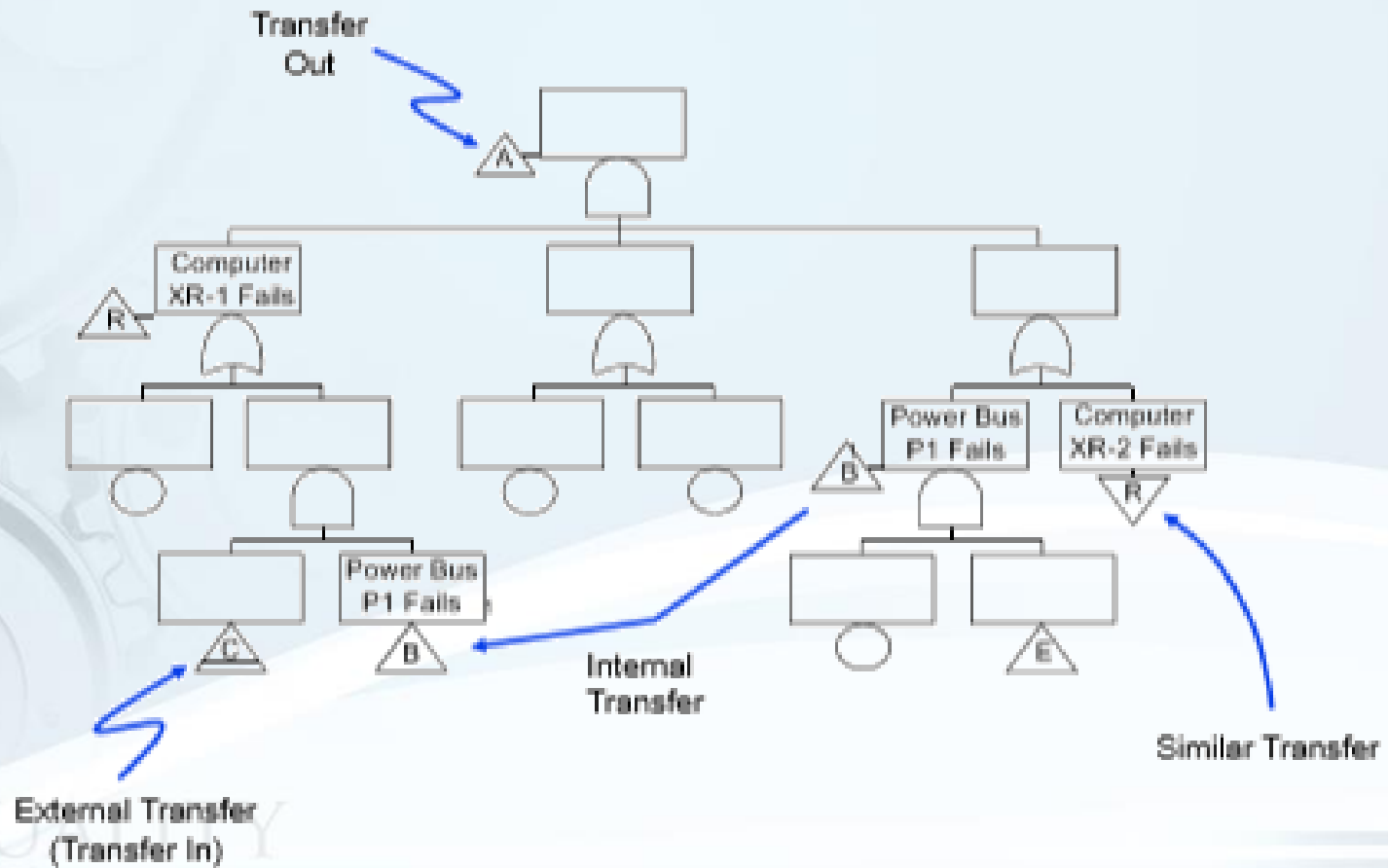
- Both C and Y1 are necessary to cause D
- Y1 is a condition or a probability
- Pass through if condition is satisfied
- Example: Ignition temperature is present, given faults cause overtemp AND probability that 700 degrees is reached

Transfer Symbol

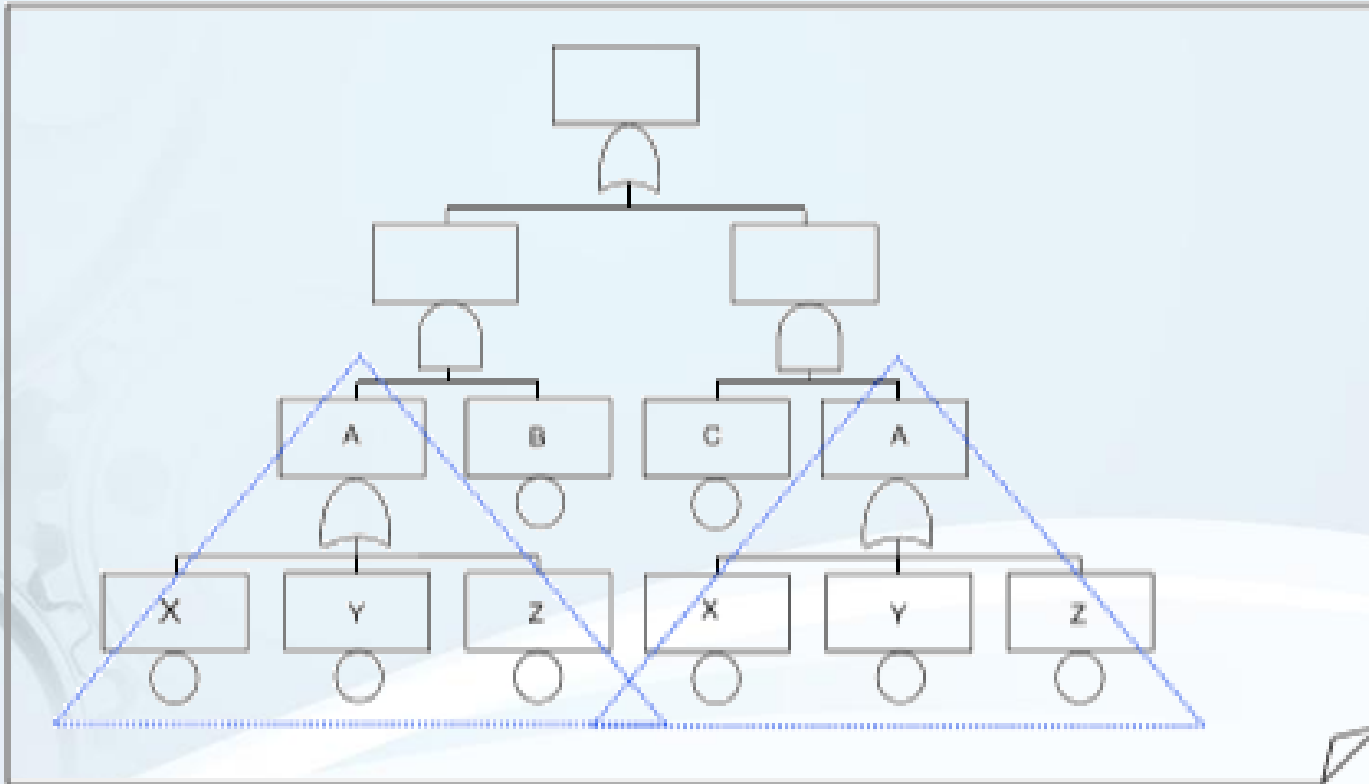
Symbol	Action	Description
	Internal Transfer	Indicates the start of a subtree branch, internal to present FT
	External Transfer	Indicates the start of a subtree branch, external to present FT
	Similar Transfer	Indicates the start of a subtree branch that is similar to another one, but with different hardware

QUALITY

Transfer Example

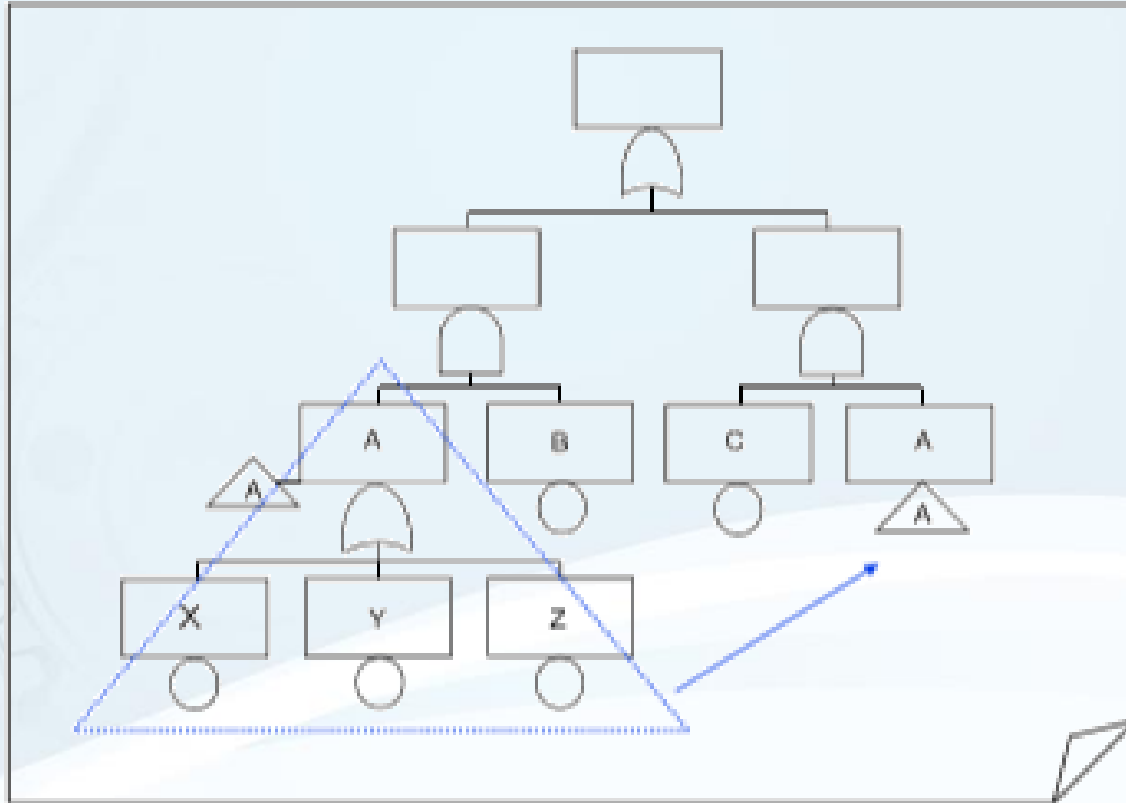


Three Methods of transfer



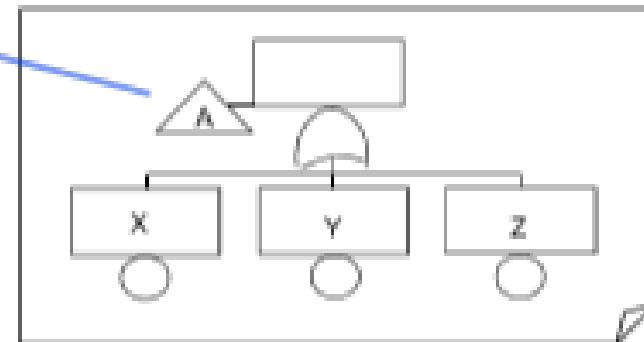
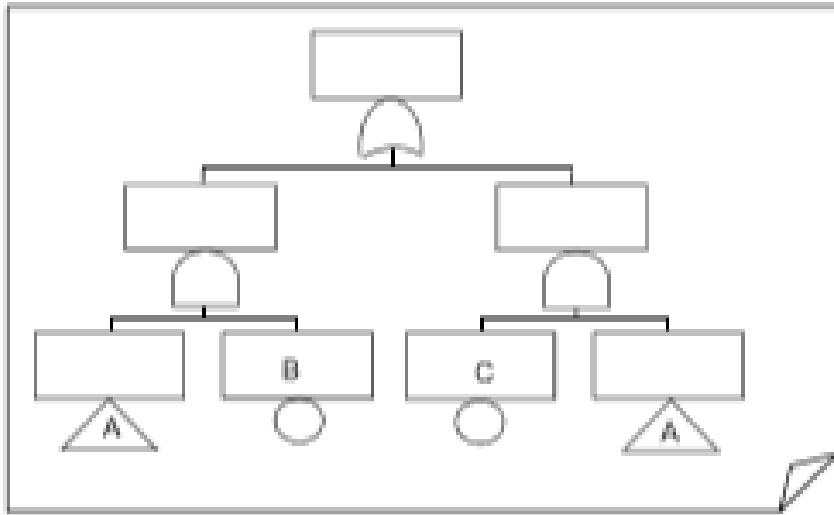
Method 1 – No Internal Transfer, MOBs on same page

Three Methods of transfer



Method 2 – Internal Transfer, MOB on same page

Three Methods of transfer



Method 3 – Internal Transfer, MOB on different page

Failure/Fault

- Failure

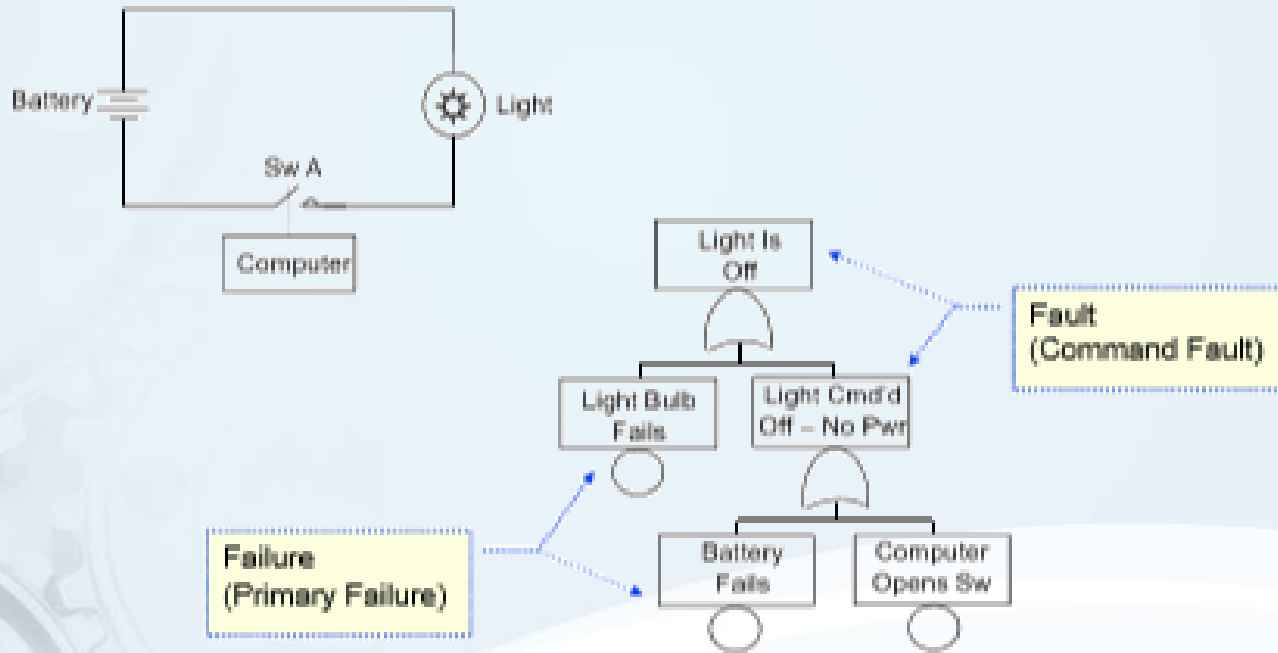
- The occurrence of a *basic component failure*.
- The result of an internal inherent failure mechanism, thereby requiring no further breakdown.
- Example - *Resistor R77 Fails in the Open Circuit Mode.*

- Fault

- The occurrence or existence of an *undesired state* for a component, subsystem or system.
- The result of a failure or chain of faults/failures; can be further broken down.
- The component operates correctly, except at the wrong time, because it was commanded to do so.
- Example – The light is failed off because the switch failed open, thereby removing power.

QUALITY

Failure/Fault Examples



All failures are faults, but not all faults are failures

QUALITY

Independent/Dependent Failure

- Independent Failure
 - Failure is not caused or contributed to by another event or component
- Dependent Failure
 - Failure is caused or contributed to by another event or component
 - A component that is caused to fail by the failure of another component
 - The two failures are directly related, and the second failure depends on the first failure occurring
 - Example - An IC fails shorted, drawing high current, resulting in resistor R77 failing open

Dependency complicates the FT mathematics

Primary Failure

- An inherent component failure mode
- Basic FT event
- A component failure that cannot be further defined at a lower level
- Example – diode inside a computer fails due to material flaw
- Symbolized by a Circle
- Has a failure rate (λ) or probability of failure



QUALITY

Secondary Failure

- A component failure that is caused by an external force to the system
- Basic FT event
- Example – Integrated circuit fails due to external RF energy
- Important factor in Common Cause Analysis
- Symbolized by a Diamond
- Has a failure rate (λ) or probability of failure

Resistor R77 fails
open from excessive
RF energy



QUALITY

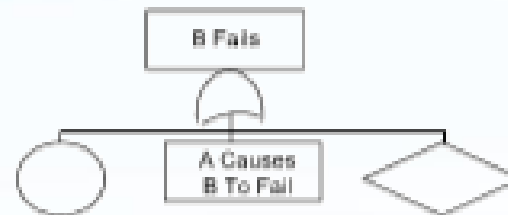
Undeveloped Failure

- A component failure that can be further defined at a lower level of detail, but is not for various reasons
 - Ground rules
 - Save analysis time and money
 - May not be a critical part of FTA
- Example – computer fails (don't care about detail of why)
- Basic FT event
- Symbolized by a Diamond
- Has a failure rate (λ) or probability of 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
- Symbolized by a gate event requiring further development

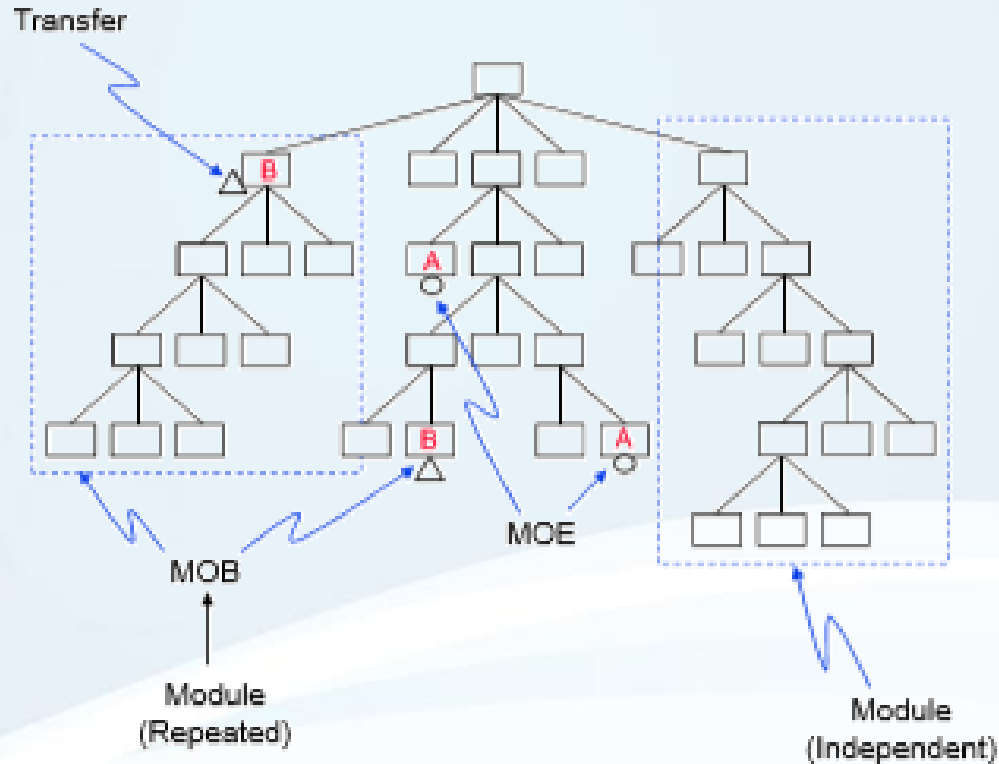


System Complexity System

- MOE
 - A Multiple Occurring Event or failure mode that occurs more than one place in the FT
 - Also known as a redundant or repeated event
- MOB
 - A multiple occurring branch (i.e., a repeated branch)
 - A tree branch that is used in more than one place in the FT
 - All of the Basic Events within the branch would actually be MOE's
- Branch
 - A subsection of the tree (subtree), similar to a limb on a real tree
- Module
 - A subtree or branch
 - An independent subtree that contains no outside MOE's or MOB's, and is not a MOB







QUALITY

MOE/MOB Example



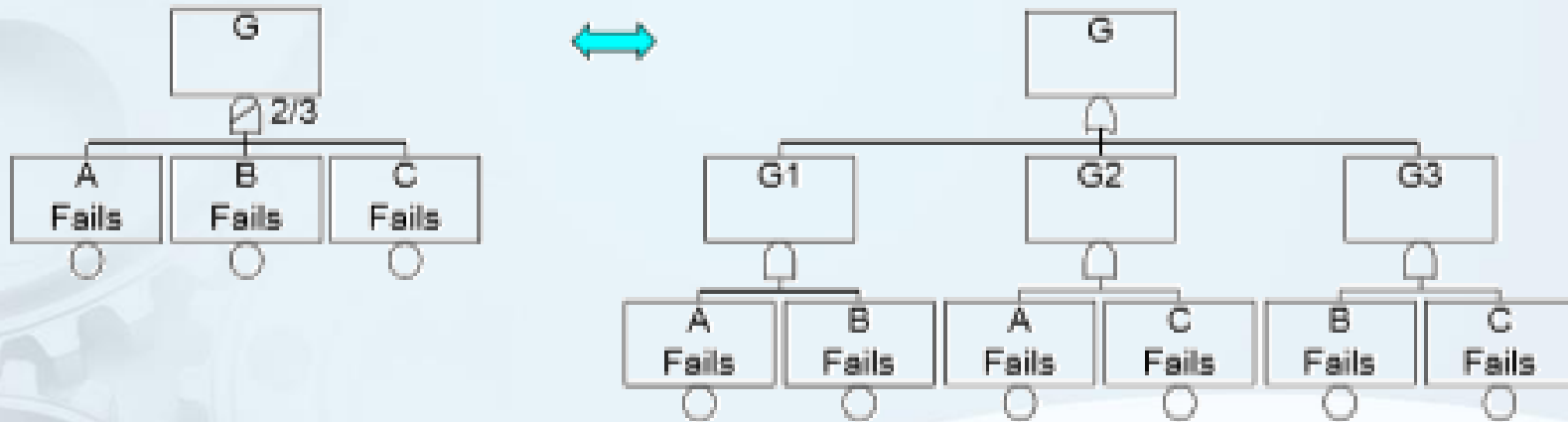
- MOE is an repeated event
- MOB is a repeated branch
- All events within an MOB are effectively MOEs

Alternate Gate Symbol

Symbol	Action	Description	Alternate Symbol
	Exclusive OR Gate	Only one of the inputs can occur, not both. Disjoint events.	
	Priority AND Gate	All inputs must occur, but in given order, from left to right.	
	M of N Gate	M of N combinations of inputs causes output to occur.	 Voting Gate

QUALITY

M/N Gate Example



- M of N gate
- Also known As Voting gate

FTA Terms/Definitions

- FT Event
 - A basic failure event on the FT
 - A normally occurring event on the FT
- FT Node
 - Any gate or event on the FT
- FT Undesired Event
 - The hazard or problem of concern for which the root cause analysis is necessary
 - The top node or event on the FT
 - The starting point for the FT analysis

QUALITY

Chapter 4

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

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?

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 Overturms 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

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

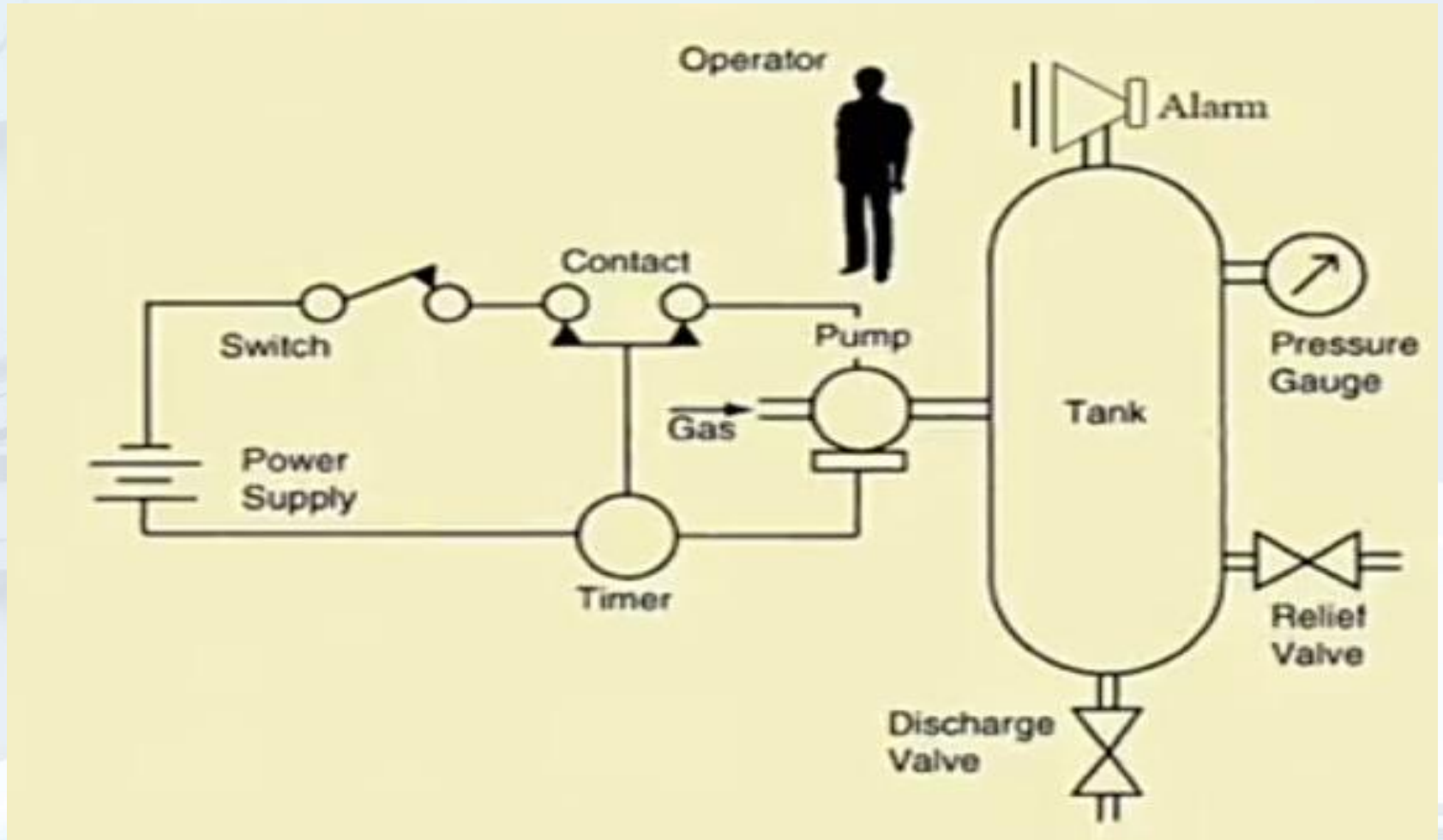
Breakout Exercise 1

Identify the Hazard (Undesired Event) &
Understanding the system

QUALITY



Pressure tank system



Some other systems for Breakout exercise

- Damping force low.
- AC not cooling.
- Axle welding crack.
- Unintended deployment of air bag.
- Seat belt failure.
- Failure of Electrical control Unit.

QUALITY



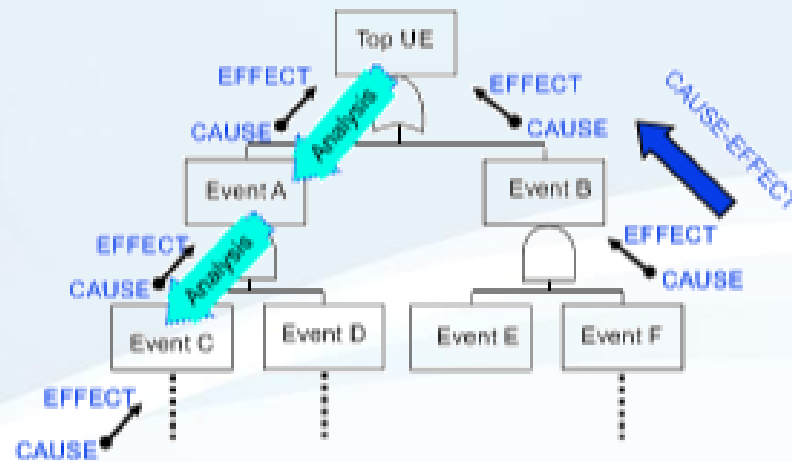
STEP 3

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



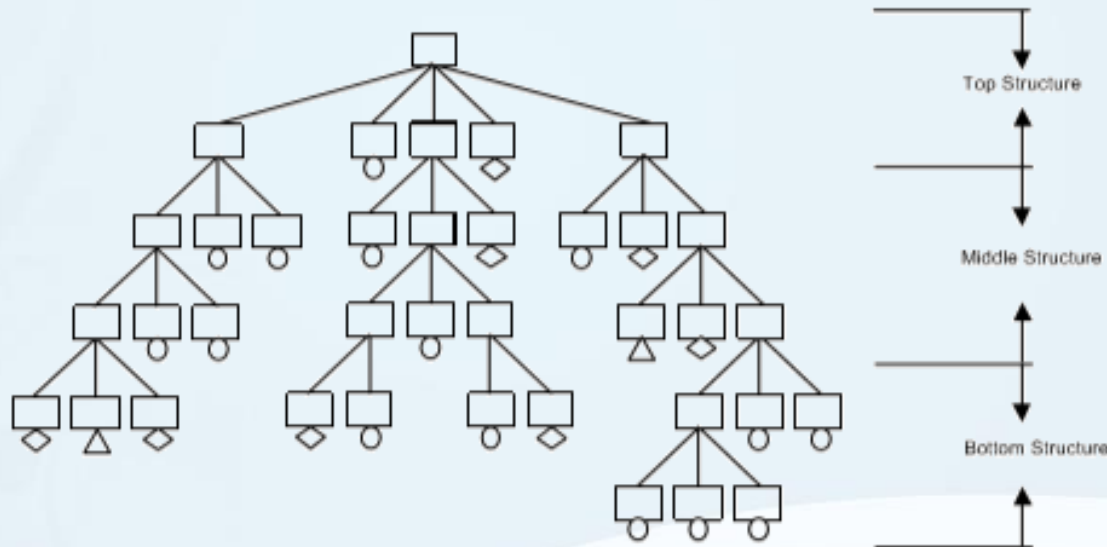
QUALITY

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

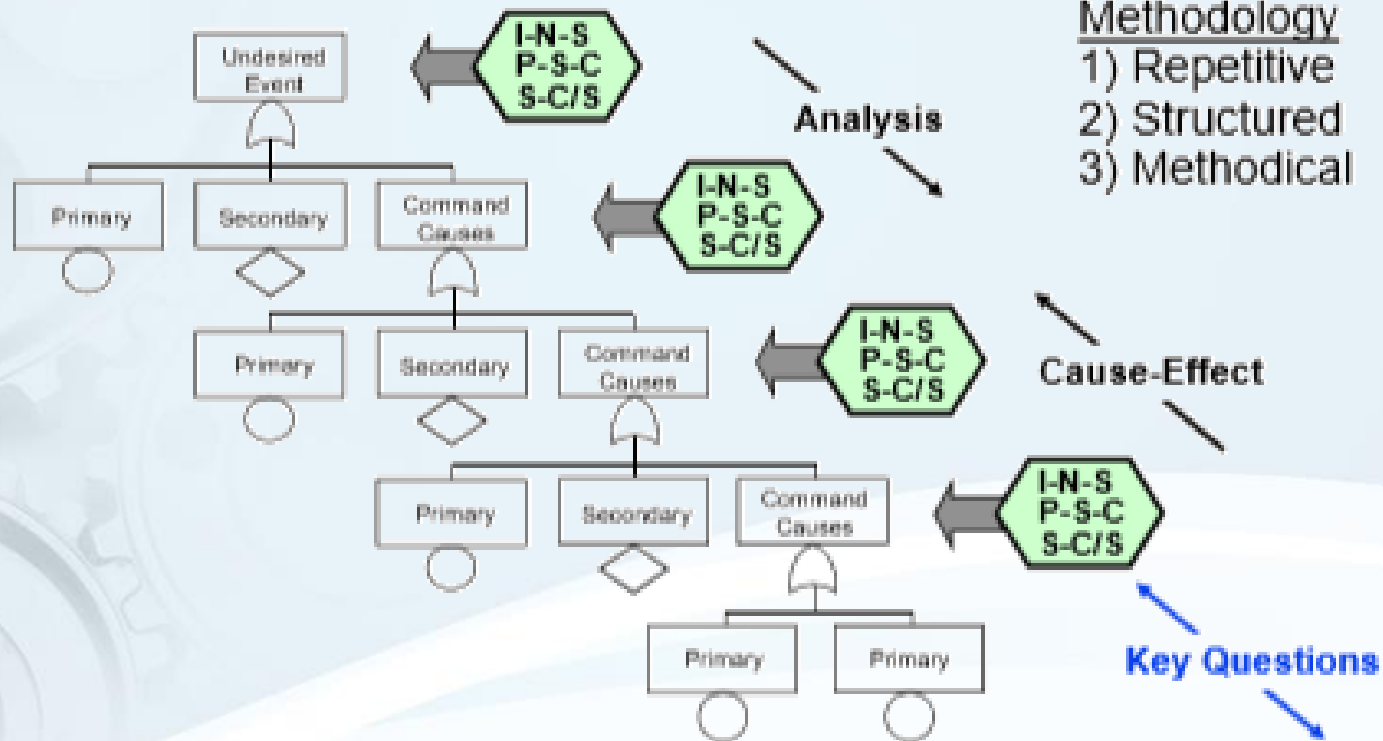
QUALITY

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



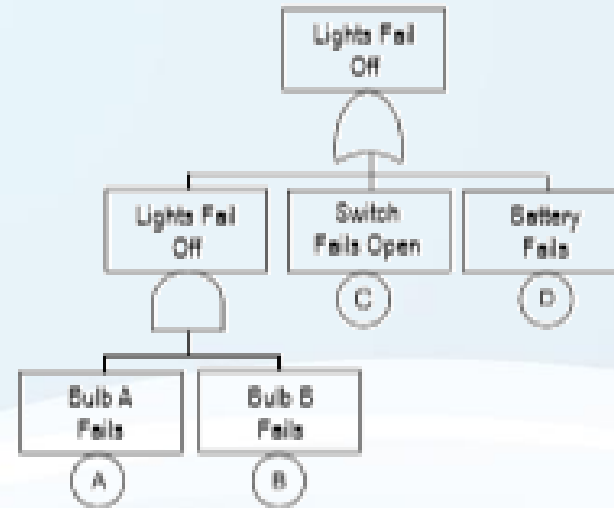
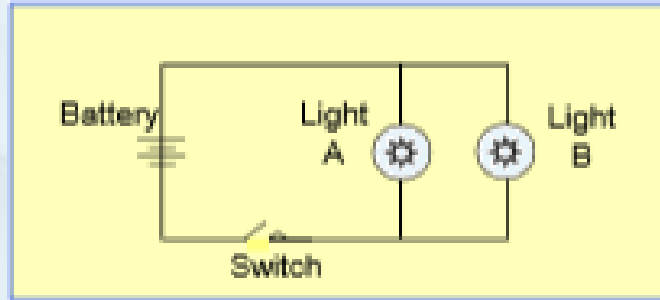
I-N-S=Immediate, Necessary, Sufficient
 P-S-C=Primary, Secondary, Command
 S-C/S=State of the Component or System

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

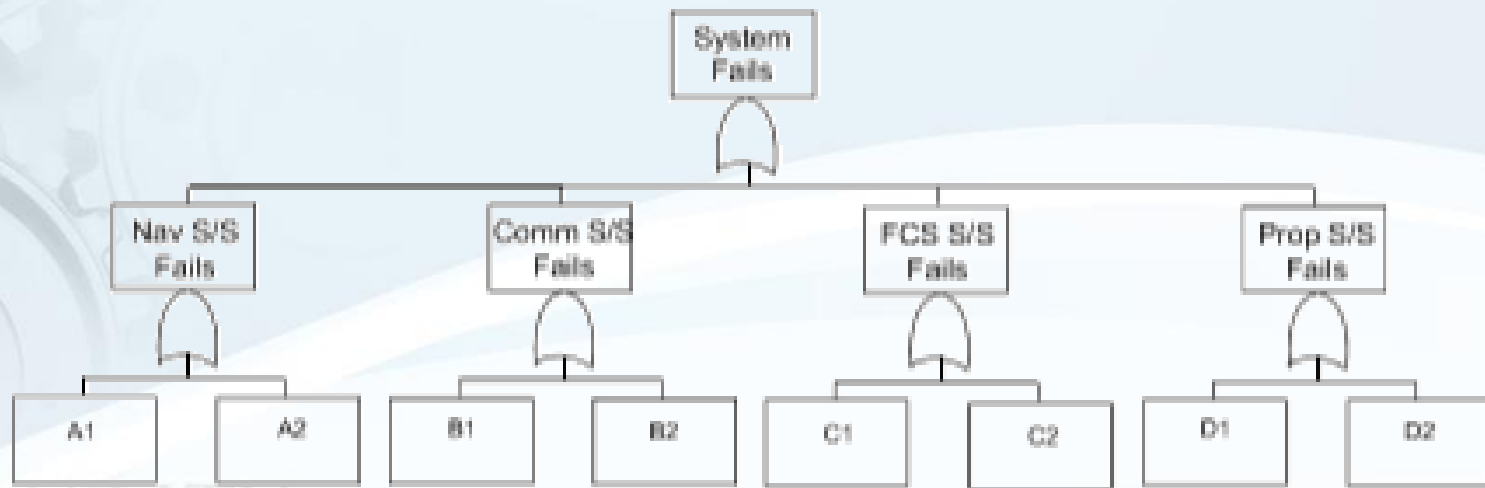
Component Approach



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

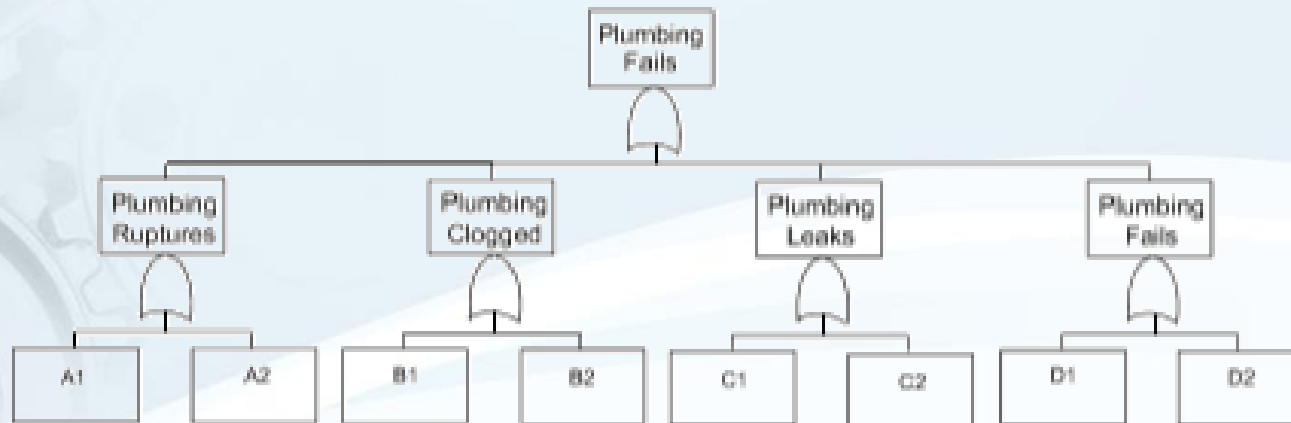
Subsystem Approach

- 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

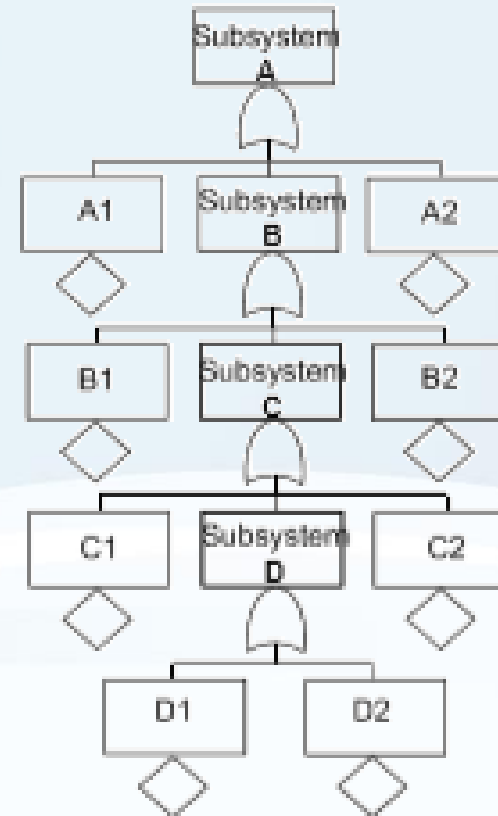
- Breakdown by Scenario
- Sometimes necessary to start large FTs
- Ignores immediate cause-effect relationship
- Eventually switch back to Functional approach
- Could be some overlap between subsystems



QUALITY

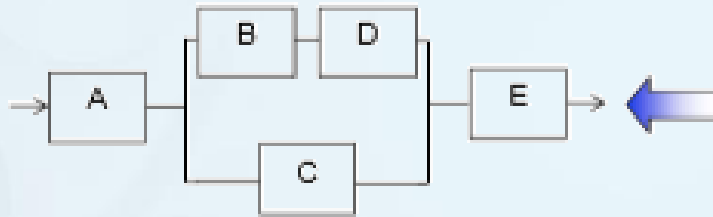
Functional Approach

- 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

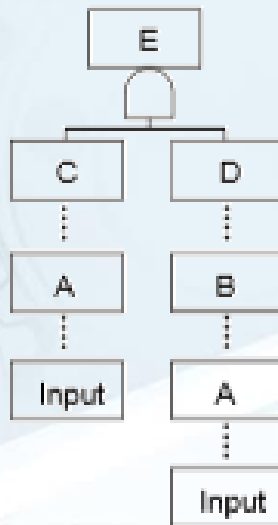


Recommended approach

Functional Approach



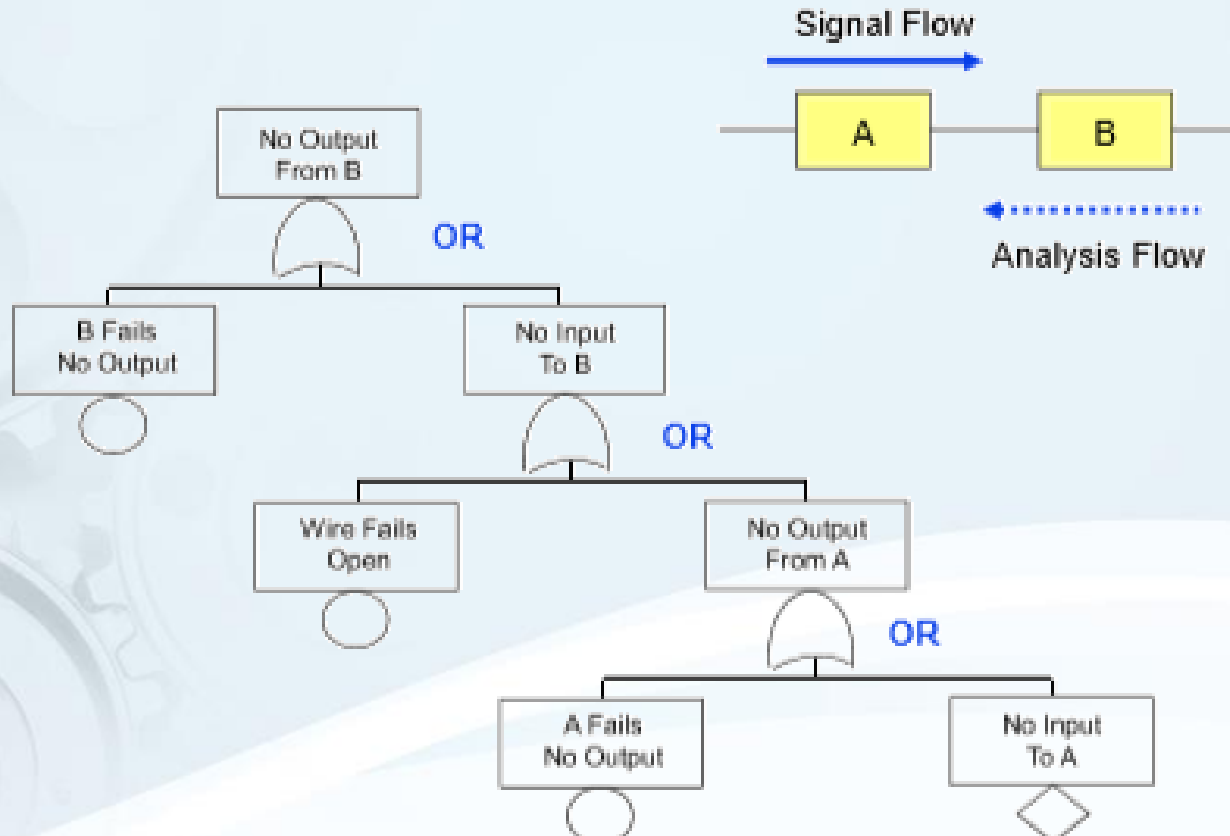
Follow the functional path



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

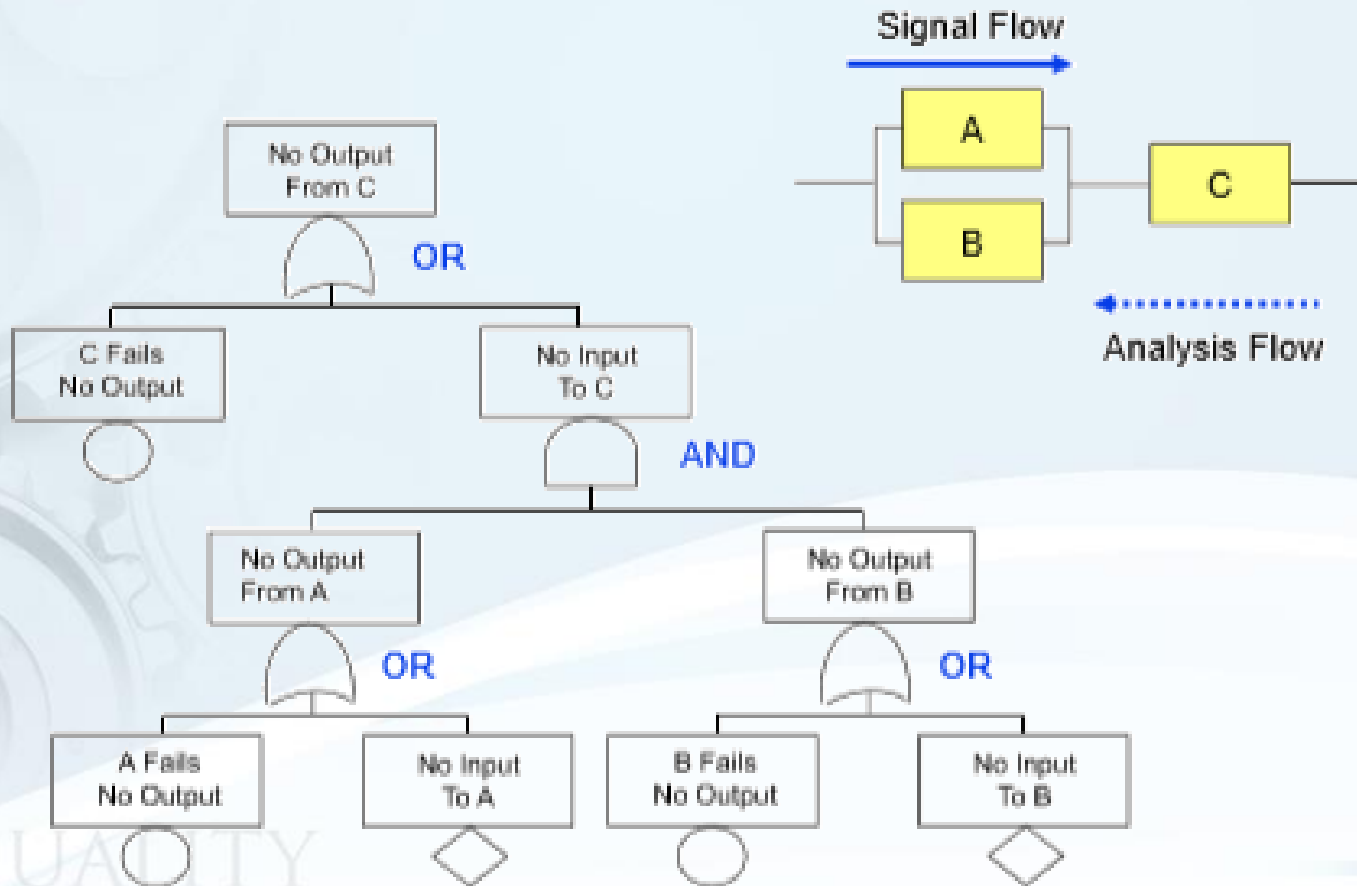
QUALITY

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

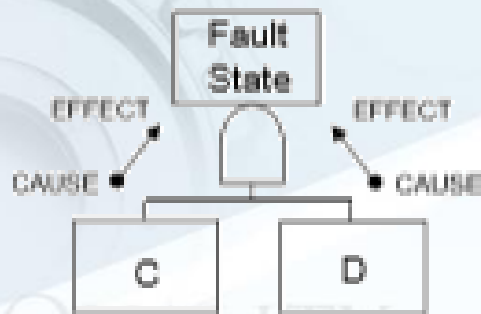
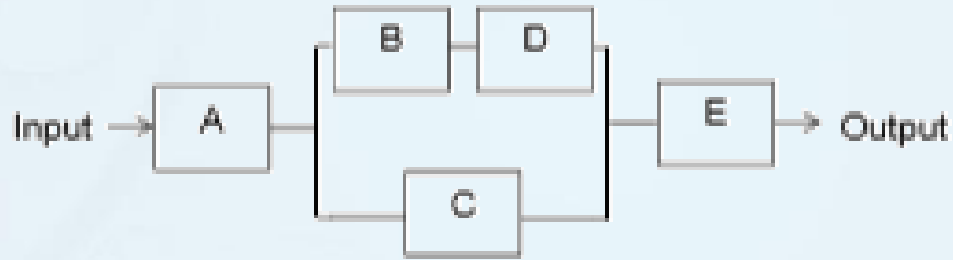
QUALITY

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



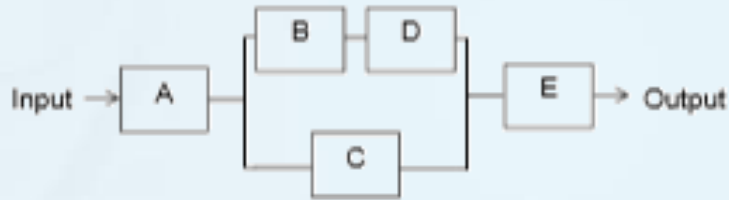
C and D are *Immediate*
C and D are *Necessary*
C and D are *Sufficient*. } To cause Fault of E

Step 2

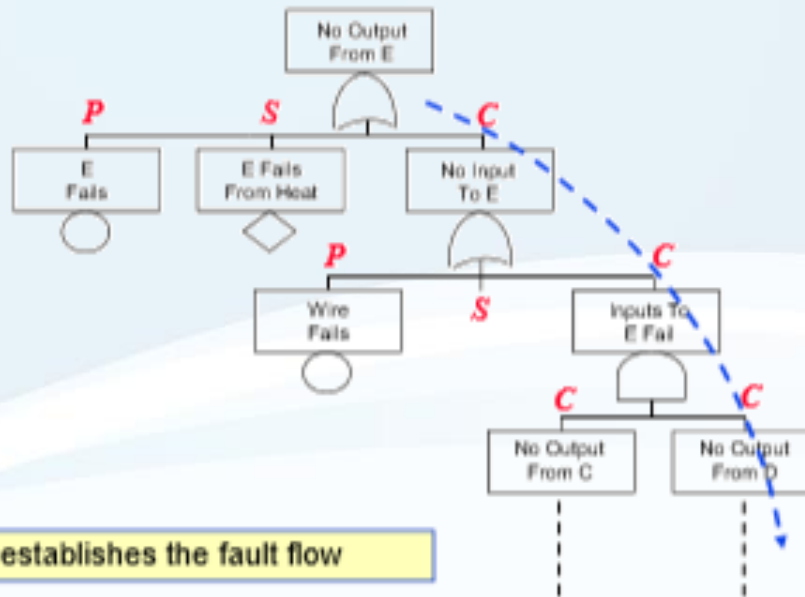
- Step 2 – *What is Primary, Secondary and Command (P-S-C) ?*
 - Read the gate event wording
 - Review I-N-S events from Step 1
 - Identify all *Primary*, *Secondary* and *Command* events causing the Gate event
 - Primary Fault – basic inherent component failure
 - Secondary Fault – failure caused by 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

Step 2



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



The Command path establishes the fault flow

QUALITY

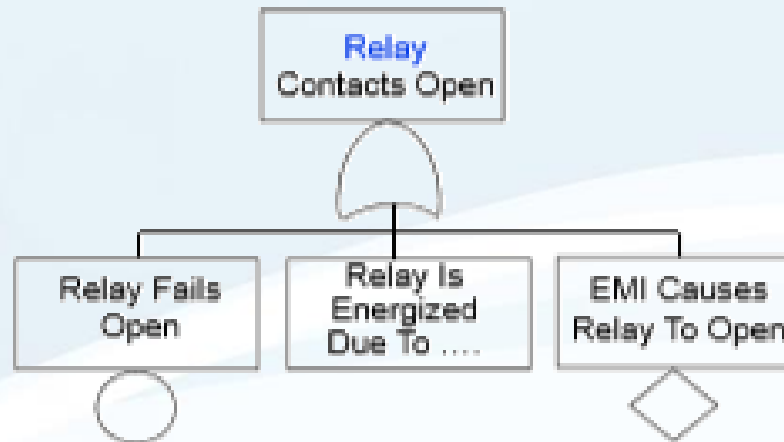
Step 3

- 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

Step3 (Cont.)

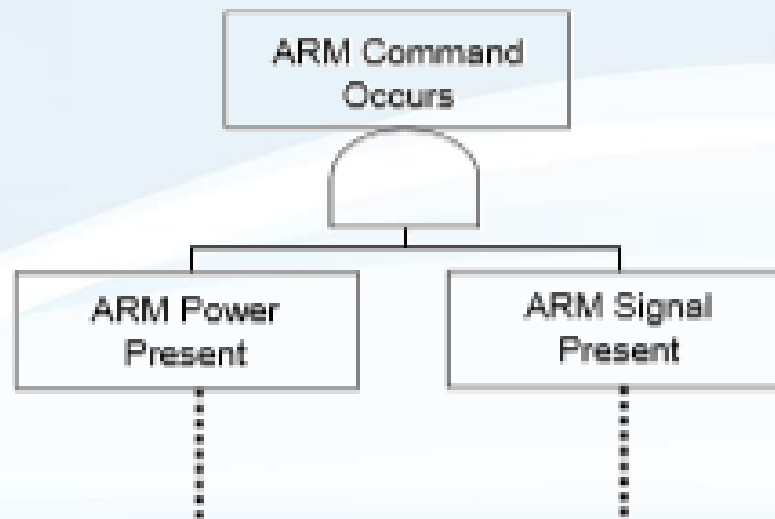
- If State of the **Component**, then:
 - Ask “what are the P-S-C causes”
 - Generally this results in an OR gate
 - If a Command event is not involved, then this branch path is complete



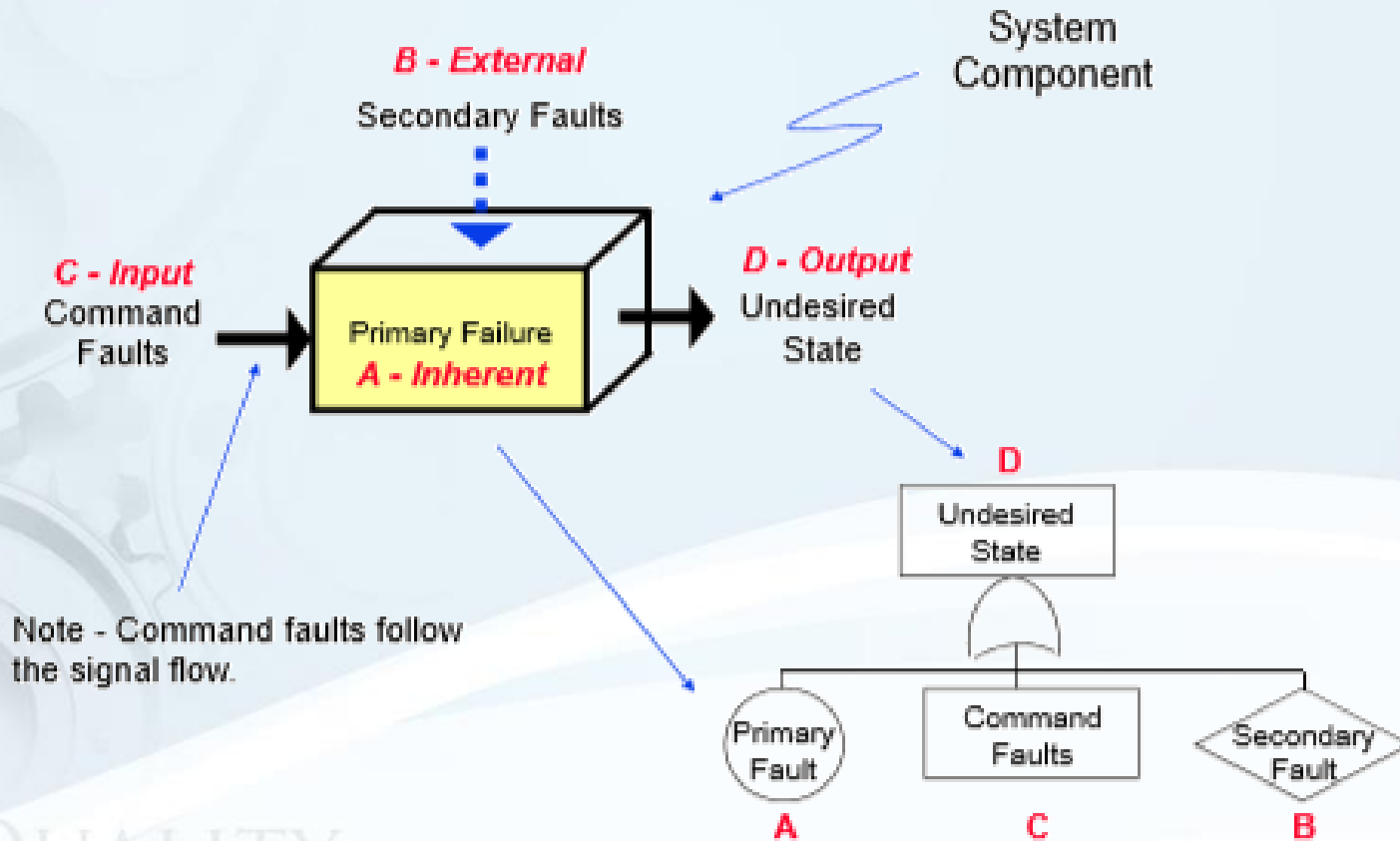
QUALITY

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

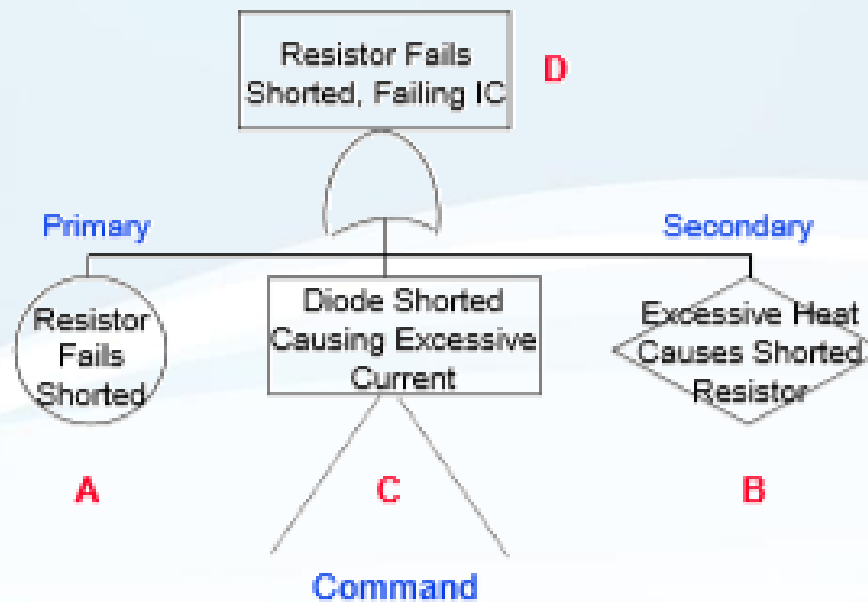
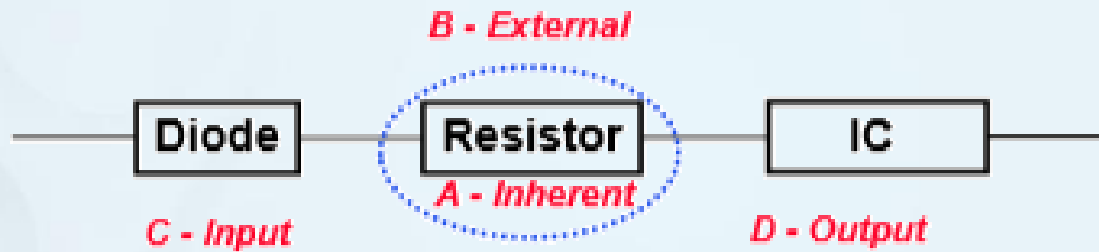


P-S-C relationship with FTA



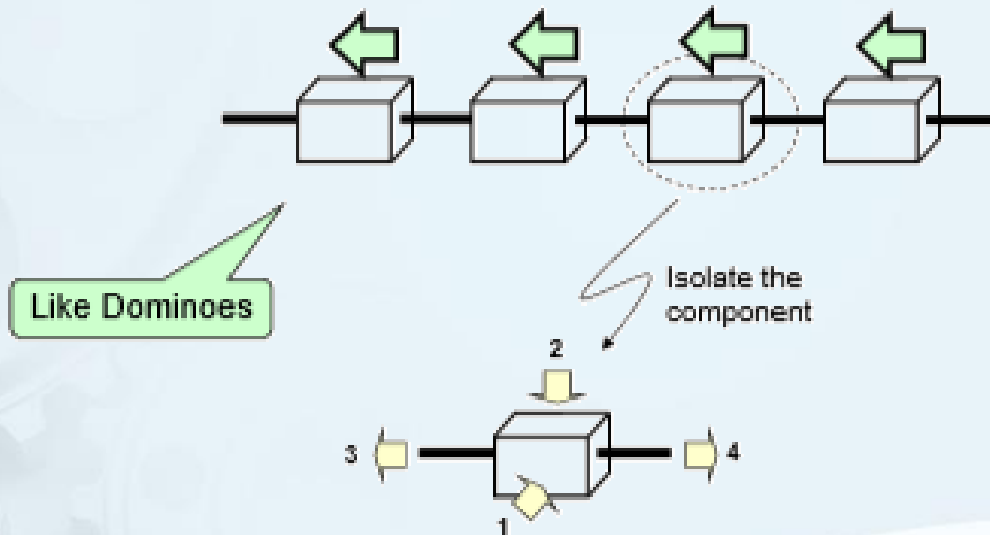
QUALITY

P-S-C Example



QUALITY

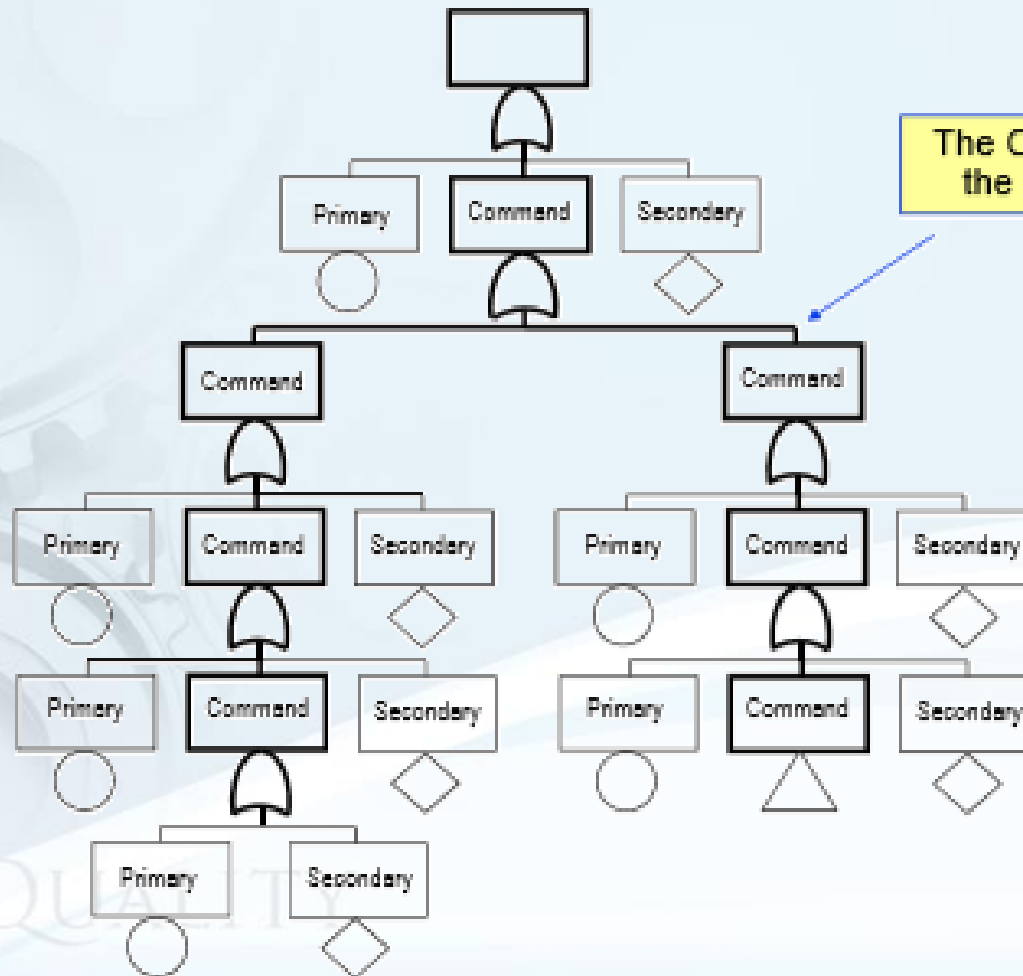
Isolate and Analyse



Analysis Views:

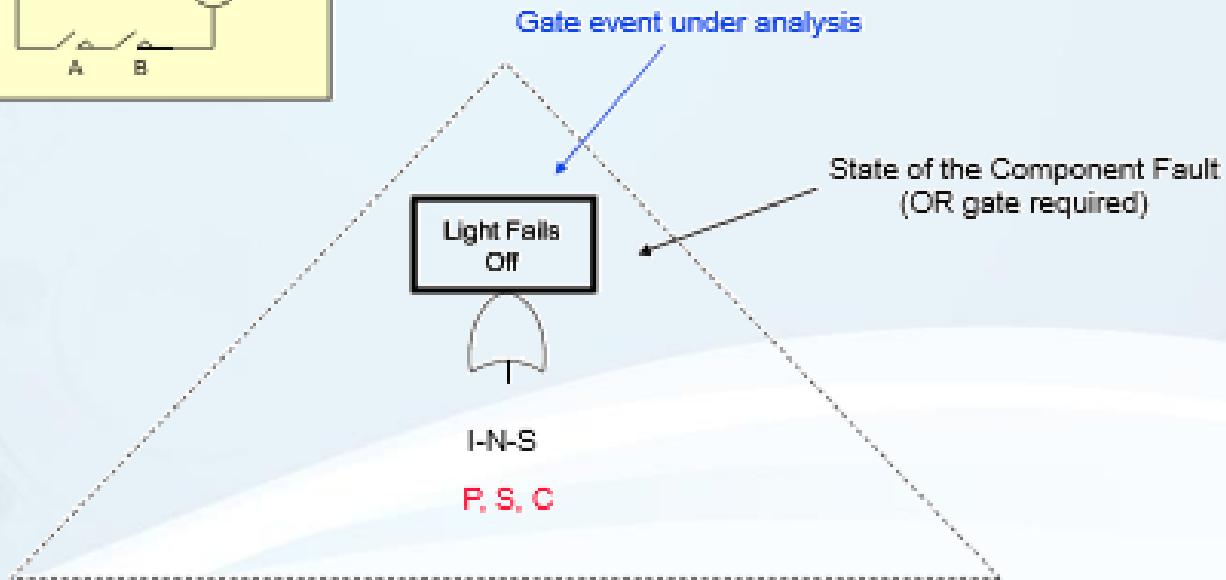
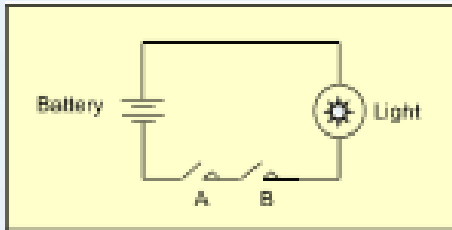
- 1) Primary - look inward
- 2) Secondary - look outward for incoming environmental concerns
- 3) Command - look backward at incoming signals
- 4) Output - look forward at possible undesired states that can be output

Example of common path



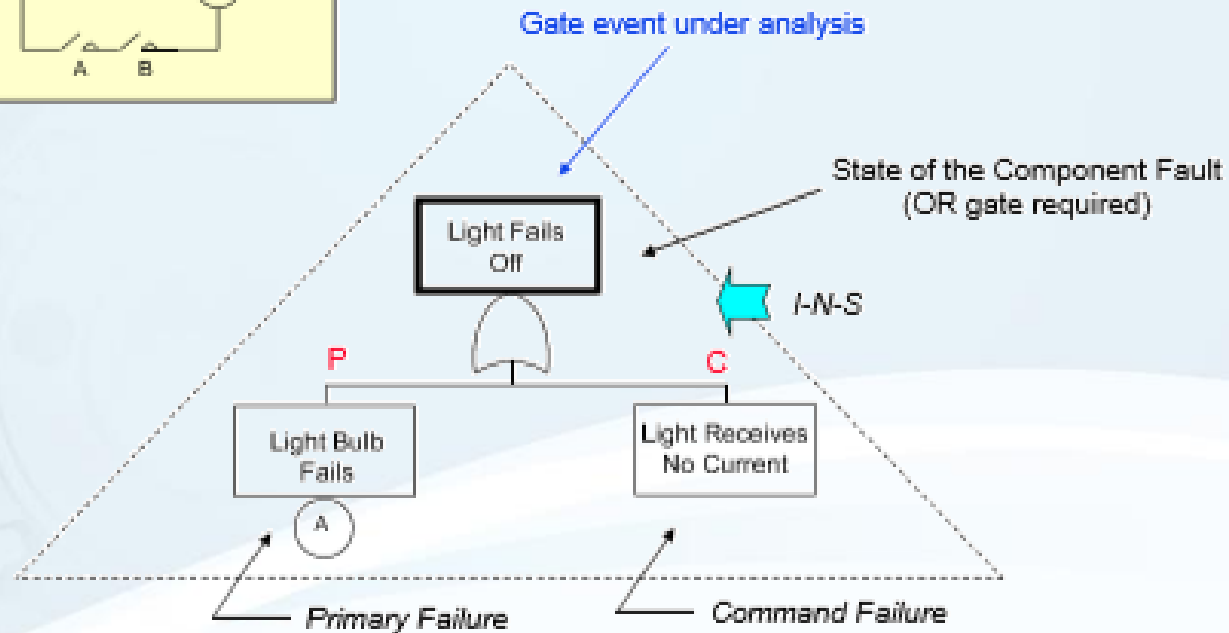
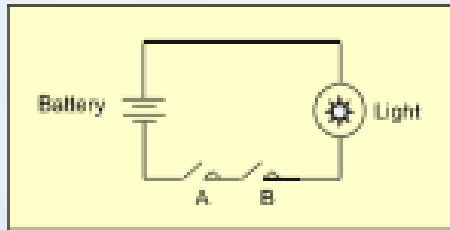
The Command path establishes the fault flow through the FT

Construction Example



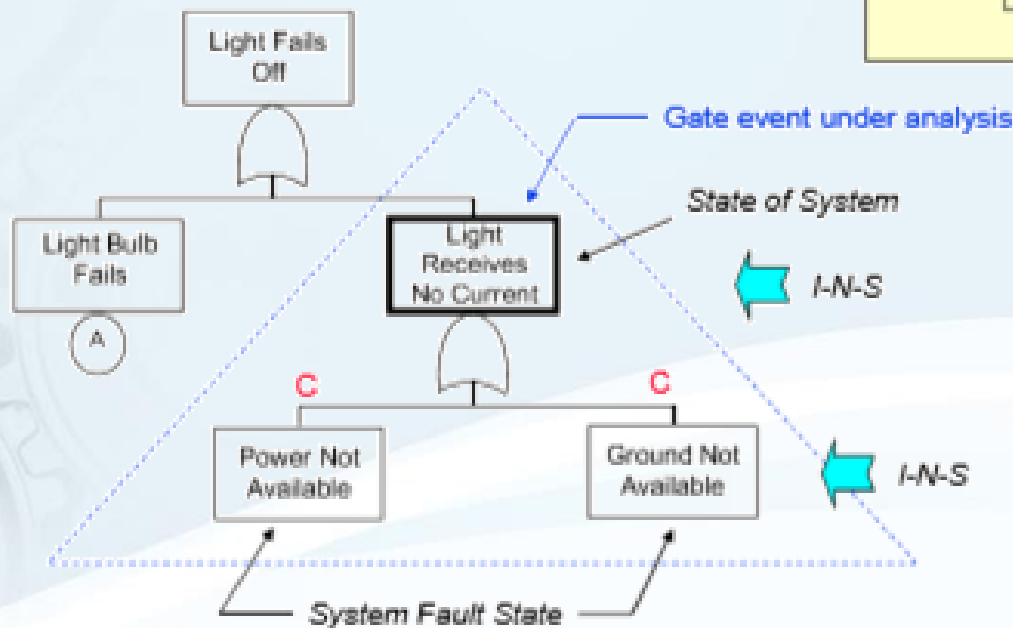
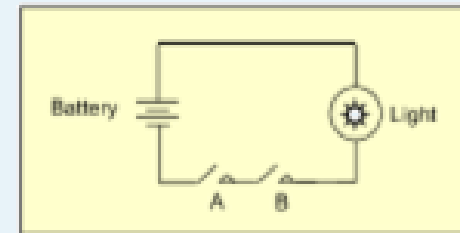
P – primary failure
S – secondary failure
C – command fault

Construction Example (Cont..)



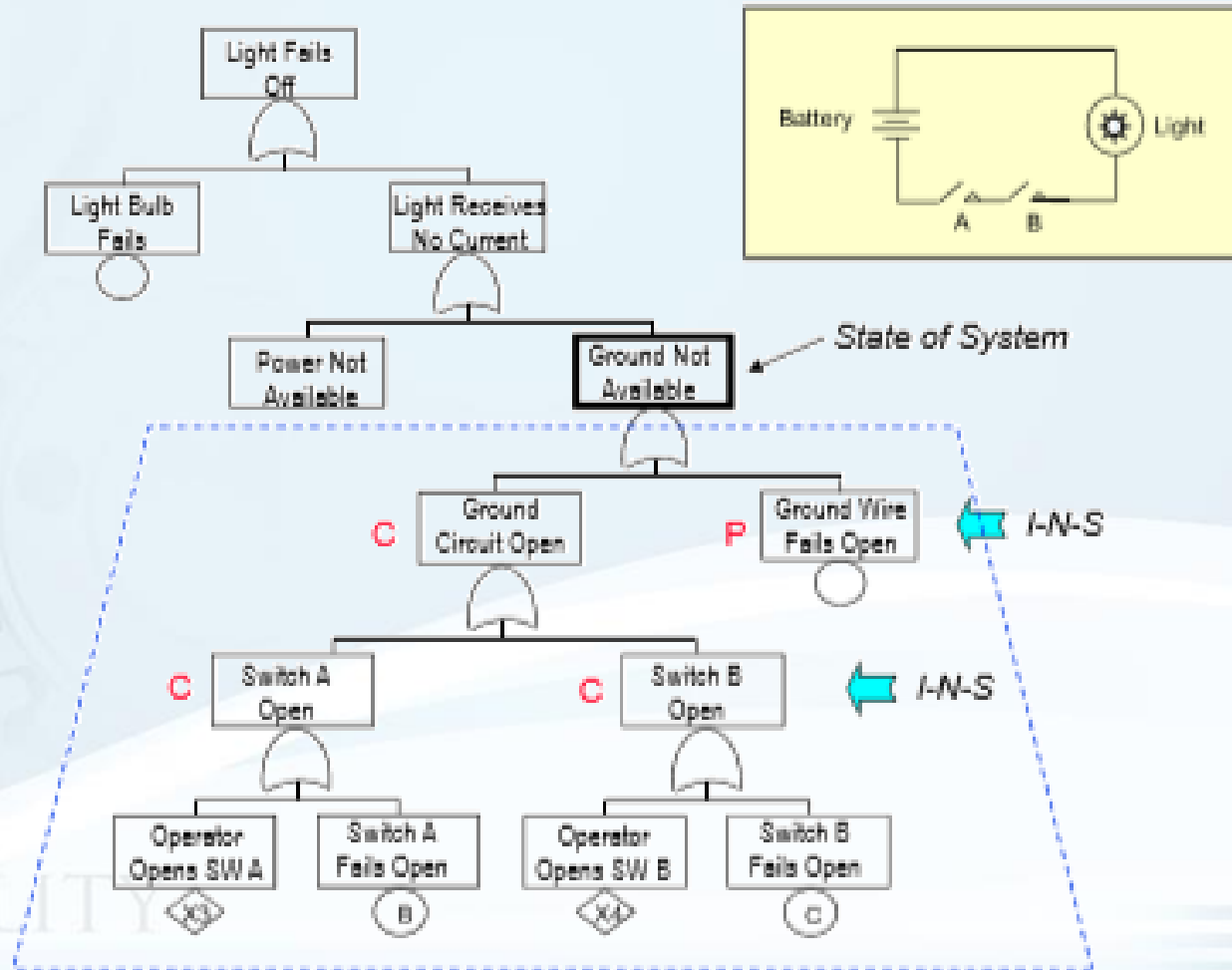
Note – This uses P-S-C, I-N-S and S-C/S

Construction Example (Cont..)

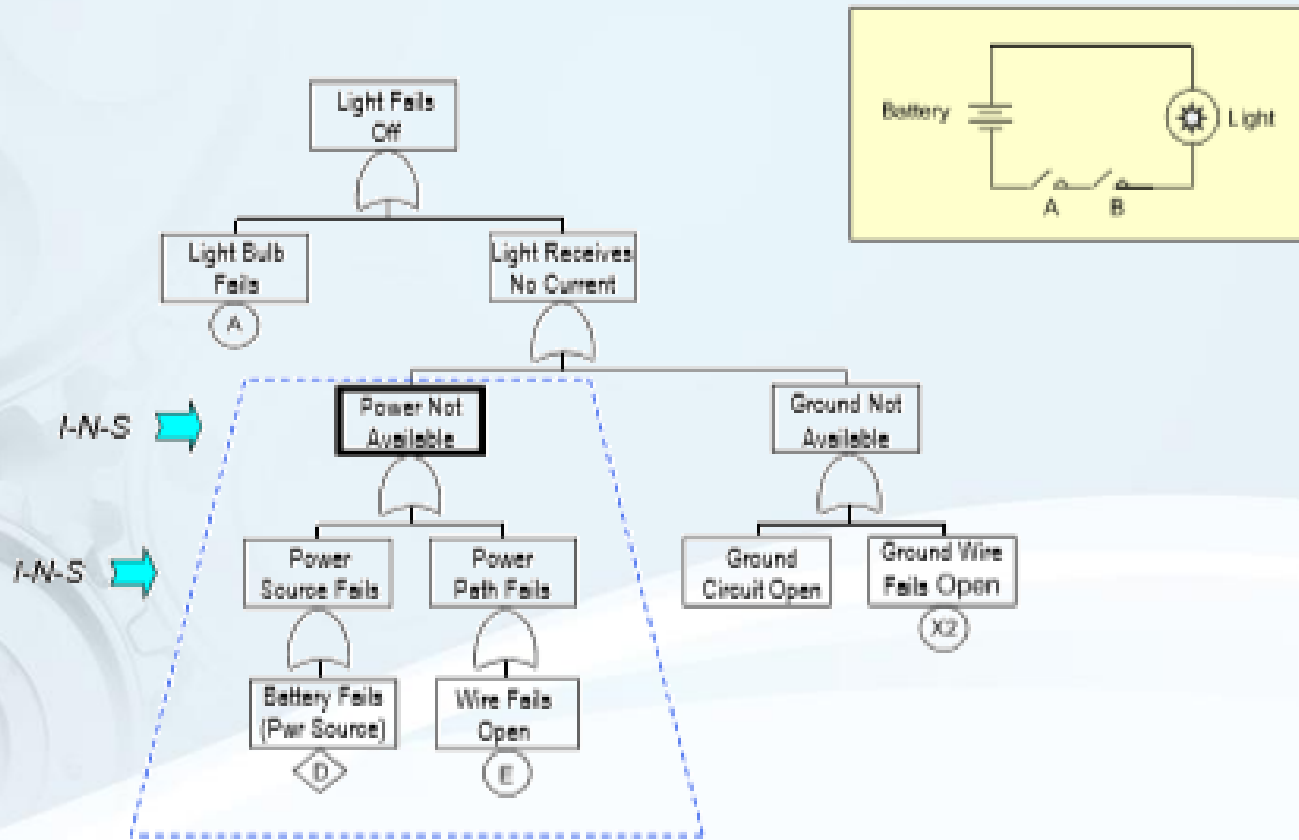


C - command fault

Construction Example (Cont..)

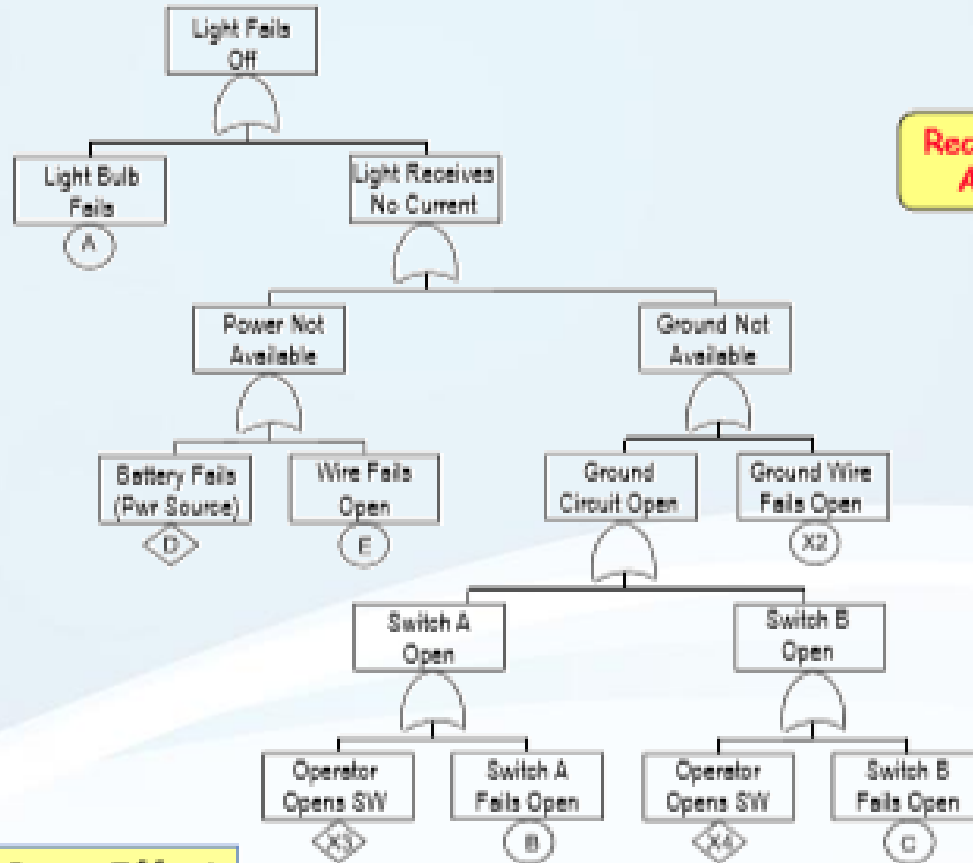


Construction Example (Cont..)



QUALITY

FTA Process – Functional Approach



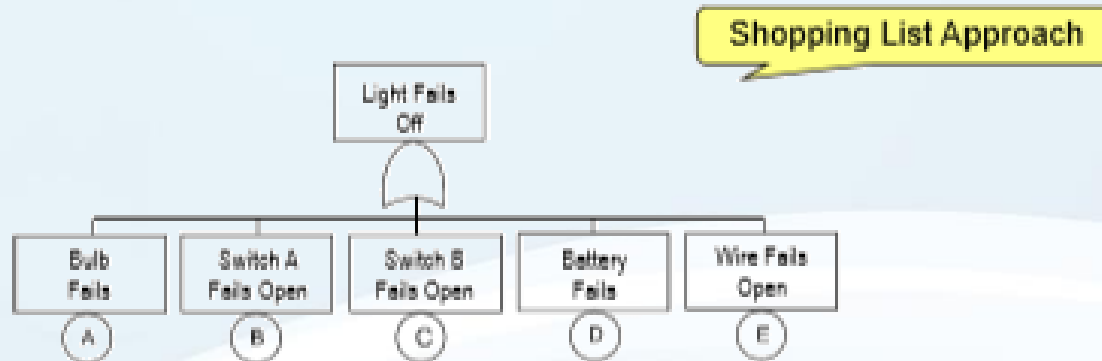
Recommended Approach

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



FTA Construction Rules

1 - Know Your System

- It is imperative to know and understand the system design and operation thoroughly
- Utilize all sources of design information
 - Drawings, procedures, block diagrams, flow diagrams, FMEAs
 - Stress analyses, failure reports, maintenance procedures
 - System interface documents
 - CONOPS
- Drawings and data must be current for current results
- Draw a Functional Diagram of the system

Rule of thumb - if you can't construct a block diagram of system you may not understand it well enough to FT

FTA Construction Rules

2 - Understand The Purpose Of Your FTA

- It's important to know why the FTA is being performed
 - To ensure adequate resources are applied
 - To ensure proper scope of analysis
 - To ensure the appropriate results are obtained
- Remember, FTA is a tool for
 - Root cause analysis
 - Identifies events contributing to an Undesired Event
 - Computes the probability of an Undesired Event
 - Measures the relative impact of a design fix
 - Logic diagrams for presentation

QUALITY

FTA Construction Rules

3 - Understand Your FT Size

- FT size impacts the entire FTA process
- As FTs grow in size many factors are affected
 - Cost (e.g., manpower)
 - Time
 - Complexity
 - Understanding
 - Traceability
 - Computation
- System factors that cause FT growth
 - System size
 - Safety criticality of system
 - System complexity
- FT factors that cause FT growth
 - MOEs and MOBs (e.g., redundancy)
 - Certain AND / OR combinations

FT size is important and has many implications

FTA Construction Rules

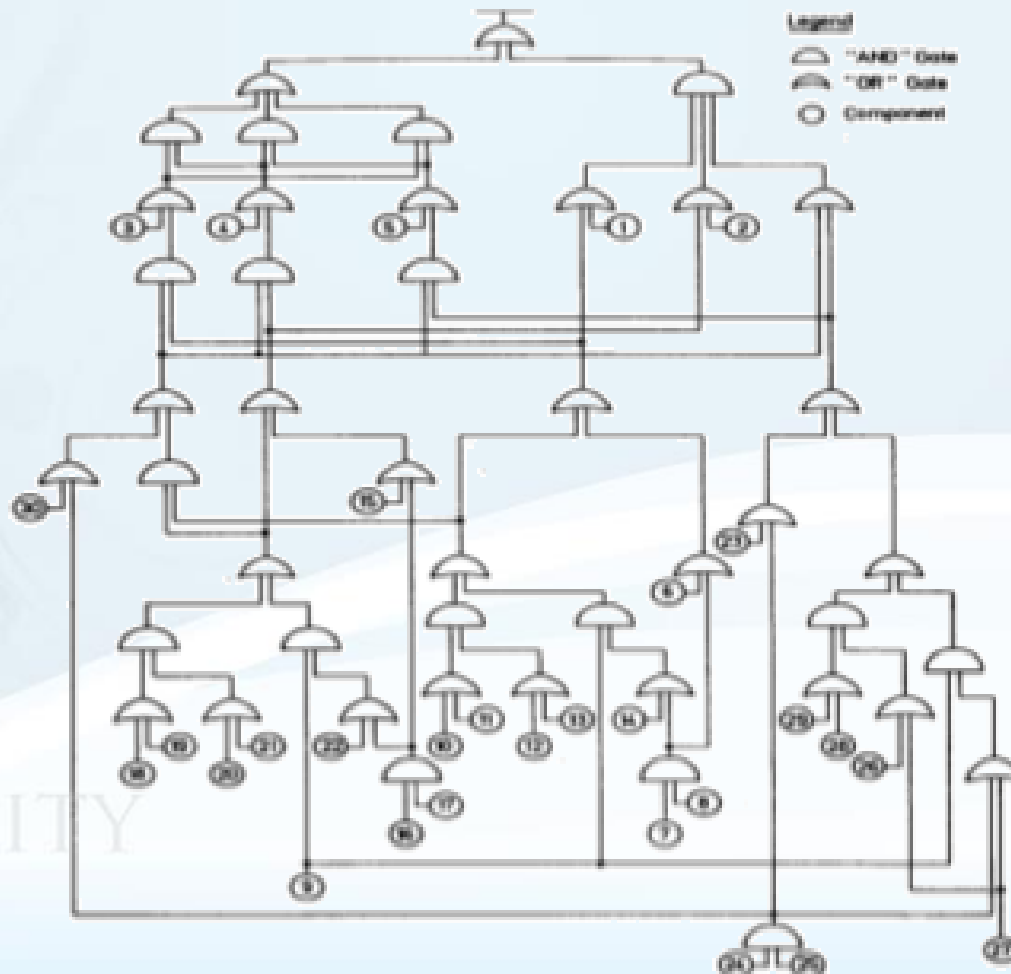
4 - Intentionally Design Your Fault Tree

- As a FT grows in size it is important develop an architecture and a set of rules
- The architecture lays out the overall FT design
 - Subsystem branches (for analysts and subcontractors)
 - Analyst responsibilities
- The rules provide consistent development guidelines
 - Ground rules for inclusion/exclusion (e.g., Human factors, CCFs)
 - Ground rules for depth of analysis (subsystem, LRU, component)
 - Ground rules for naming conventions (component types, MOEs)
 - Ground rules for component database

Foresight helps avoid future problems

FTA Construction Rules

Don't Do This! -- Plan Ahead



FTA Construction Rules

5 - Ensure the FT is Correct and Complete

- FT completeness is critical
 - Anything left out of the FTA skews the answer
 - The final result will only reflect what was included in the FT
 - The FTA is not complete until all root causes have been identified
- FT correctness is critical
 - If the FT is not correct the results will not be accurate
- Conduct FT peer review to ensure completeness/correctness
 - Involve other FT experts
 - Involve system designers
- Items often overlooked in FTA
 - Human error
 - Common cause failures
 - Software factors (design may have dependencies)
 - Components or subsystems considered not applicable

FT results are skewed if the FT is not complete and correct

FTA Construction Rules

6 - Know Your Fault Tree Tools

- Know basic FT tool capabilities
 - Construction, editing, plotting, reports, cut set evaluation
- Know FT tool user friendliness
 - Intuitive operation
 - Easy to use and remember
 - Changes are easy to implement
- Single vs. multi-phase FT
- Qualitative vs. quantitative evaluation
- Simulation vs. analytical evaluation (considerations include size, accuracy, phasing)

QUALITY

FTA Construction Rules

Tools (continued)

- Know FT tool limitations
 - Tree size (i.e., max number of events)
 - Cut set size
 - Plot size
- Understand approximations and cutoff methods, some can cause errors
- Gate probabilities could be incorrect when MOEs are involved
- Test the tool; don't assume answers are always correct

Don't place complete trust in a FT program

FTA Construction Rules

7 - Understand Your FTA Results

- Verify that the FTA goals were achieved
 - Was the analysis objective achieved
 - Are the results meaningful
 - Was FTA the right tool
 - Are adjustments necessary

- Make reasonableness tests to verify the results
 - Are the results correct
 - Look for analysis errors (logic, data, model, computer results)
 - Are CSs credible and relevant (if not revise tree)
 - Take nothing for granted from the computer
 - Test your results via manual calculations

FTA Construction Rules

8 - Document Your FTA

- Formally document 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
- May support future questions or analyses

Documentation is essential

QUALITY

FTA Construction Rules

Documentation (continued)

- Provide complete documentation
 - Problem statement
 - Definitions
 - Ground rules
 - References
 - Comprehensive system description
 - Data and sources (drawings, failure rates, etc.)
 - FT diagrams
 - FT tree metrics
 - FT computer tool description
 - Results
 - Conclusions

Document the number of hours to perform the FTA for future estimates

FTA Construction Rules

9 - Think in Terms of Failure Space

- Remember, it's a "fault" tree, not a "success" tree
 - Analysis of failures, faults, errors and bad designs
- No magic
 - Do not draw the fault tree assuming the system can be saved by a miraculous failure
 - This is normally referred to as the "No Magic Rule"
- No operator saves
 - When constructing FT logic do not assume that operator action will save the system from fault conditions
 - Only built-in safety features can be considered
 - Operator errors can be considered in the FT, but not operator saves
 - The system design is under investigation, not the operator performing miracles

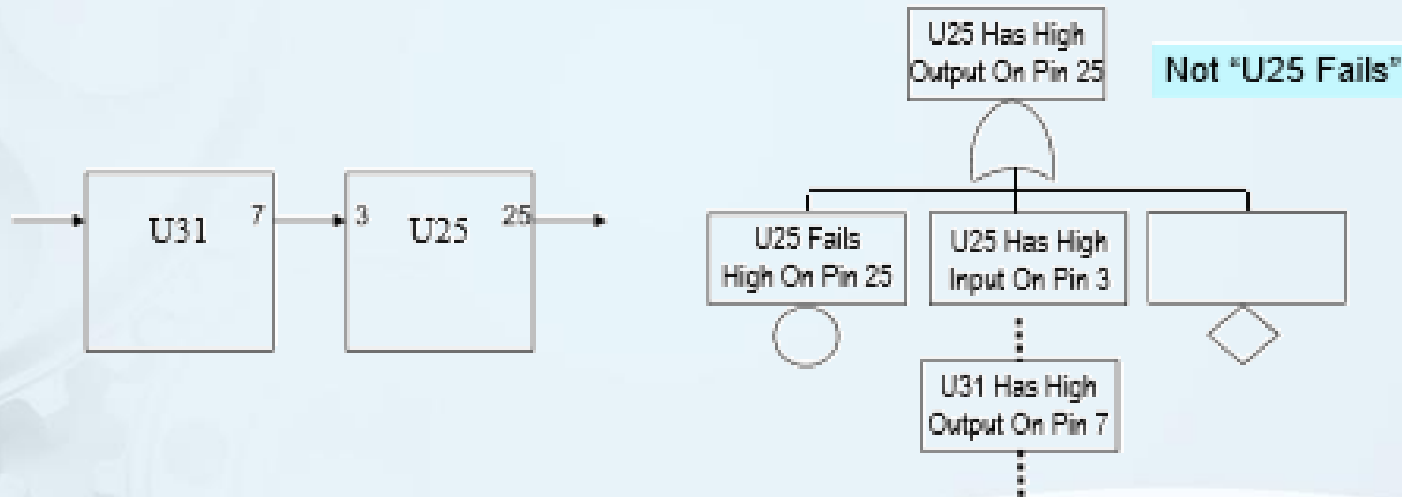
FTA Construction Rules

10 - Correct Node Wording Is Important

- Be clear and precise
- Express fault event in terms of
 - Device transition
 - Input or output state
- Be very descriptive in writing event text
 - "Power supply fails" vs. "Power supply does not provide +5 VDC"
 - "Valve fails in closed position" vs. "Valve fails"
- Do not
 - Use the terms Primary, Secondary or Command
 - ◆ Thought process
 - ◆ Symbols already show it
 - Use terms Failure or Fault (if possible) – not enough information

Good node wording guides the analysis process

FTA Construction Rules



Proper wording enhances the logic process

QUALITY

FTA Construction Rules

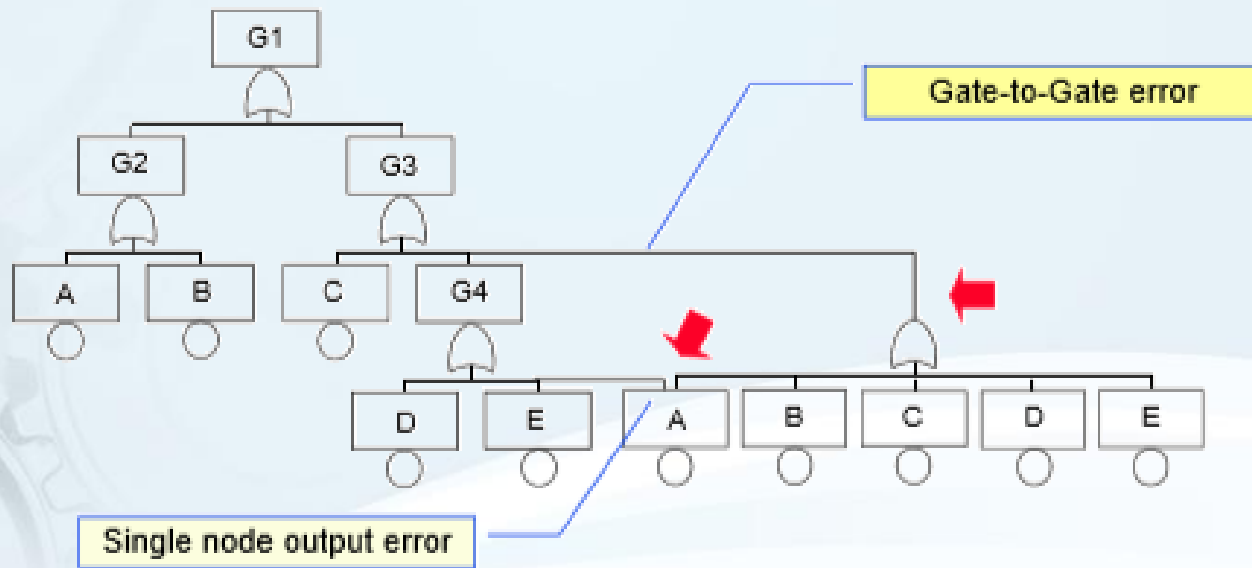
11 - Follow Standard Construction Rules

- No gate-to-gate diagrams
 - Do not draw a gate without a gate node box and associated descriptive text and rectangle
- Use only one output from a node
 - Do not connect the output of a node to more than one input nodes.
 - Some analysts attempt to show redundancy this way, but it becomes cluttered and confusing.
 - Most computer codes cannot handle this situation anyway.

QUALITY

FTA Construction Rules

Construction Errors



Usually not possible with computer FT programs

FTA Construction Rules

FT Construction Rules (cont'd)

- Construct the FT to most accurately reflect the system design and logic
 - Do not try to modify the tree structure to resolve an MOE.
 - Let the FT computer software handle all MOE resolutions.
- Keep single input OR gates to a minimum
 - When the words in a Node box exceed the box limit, you can create another input with a Node box directly below just to continue the words
 - Use the Notes if additional words are needed. Its okay to do but prudence is also necessary
- Use House events carefully
 - A House (Normal event) never goes into an OR gate, except in special cases, such as a multi-phase simulation FT

QUALITY

FTA Construction Rules

FT Construction Rules (cont'd)

- Do not label fault events on the tree as *Primary*, *Secondary* and *Command* failures
 - Go into detail and be descriptive. These terms are more for the thought process than the labeling process.
- When possible add traceability detail
 - Put drawing numbers and part numbers in the fault event or in the notes.
 - This provides better traceability when the tree is being reviewed or checked, or when the tree is being modified after a lengthy time period.

QUALITY

FTA Construction Rules

FT Construction Rules (cont'd)

- Operator error should be included in the analysis where appropriate
 - It is up to the analyst and the purpose/objective of the FTA as to whether the event should be included in quantitative evaluations
 - The decision needs to be documented in the analysis ground rules

- Take a second look at all tree logic structure
 - Sometimes what appears to be a simple and correct tree logic structure might actually be flawed for various reasons
 - ◆ Example -- mutually exclusive events, logic loops, etc.
 - Make sure there are no leaps or gaps in logic
 - The tree structure may need revising in these cases

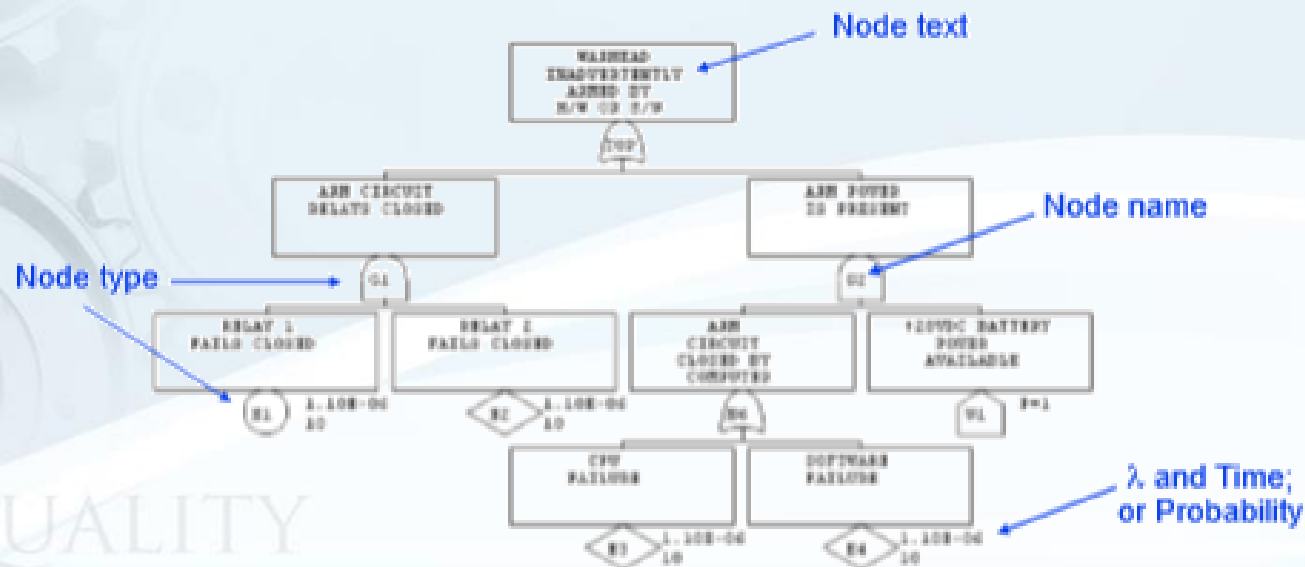
QUALITY

FTA Construction Rules

12 - Provide Necessary Node Data

- Node name
- Node text
- Node type
- Basic event probability (for quantification only)

Four items are essential



FTA Construction Rules

13 - Apply FT Aesthetics

- When the FT structure looks good it will be better accepted
- A level FT structure looks best
 - No zig-zags
- Balance page breaks & FT structure
 - Avoid too little info on a page (i.e., 2 or 3 events)
- Always use standard FT symbols (defined in NUREG book)
- Computerized construction tools provides better graphics than manual methods

QUALI A level and balanced FT structure is easier to read

FTA Construction Rules

14 - Computerized Evaluation Is Essential

- FT quantification is easy when the FT is small and simple
 - Manual calculations are easy
- FT quantification is difficult when the FT is large and complex
 - Manual quantification becomes too difficult without errors
- Hand drawn FTs typically have more errors

QUALITY

FTA Construction Rules

15 - Validate all CSs

- CSs are very important
 - They show where to fix system (weak design points)
 - They show the importance of specific components
 - They are necessary for most numerical calculations
- Always verify that all CSs are valid
 - If they are not right the FT is incorrect

QUALITY

FTA Construction Rules

16 - Perform a Numerical Reality Check

- Never completely trust the results of a computer program
 - Some algorithms may have errors
 - Proprietary approximations may not always work
- Perform a rough calculation manually to check on the computer results
- A large deviation could indicate a problem

QUALITY

FTA Construction Rules

17 - Verify All MOEs and MOBs

- Review MOEs very carefully
 - Their effect can be important - common cause, zonal analysis
 - They can cause large numerical error (or none at all)
 - They can hide or emphasize redundancy
- An MOE or MOB can be inadvertently created by erroneously using the same event name twice

QUALITY

FTA Construction Rules

18 - FTs Are Only 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 (2.0×10^{-8} is as good as 1.742135×10^{-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)

QUALITY

FTA Construction Rules

19 - 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)
- Proven data is best
- Don't be afraid of raw data
 - Data estimates can be used
 - Useful for rough estimate
 - Results must be understood

Even raw data provides useful results

QUALITY

FTA Construction Rules

20 – Always Provide Data Sources

- MIL-HDBK-217 Electronic Parts Predictions
- Maintenance records
- Vendor data
- Testing
- Historical databases

QUALITY

FTA Construction Rules

21 – The Human Is A System Element

- The human is often a key element in the system lifecycle
 - Manufacturing, assembly, installation, operation, decommissioning
- The human might be the most complex system element
- Human error includes
 - Fails to perform function (error of omission)
 - Performs incorrectly
 - Performs inadvertently (error of commission)
 - Performs wrong function
- Human error can
 - Initiate a system failure or accident
 - Fail to correctly mitigate the effects of a failure (e.g., ignored warning lights)
 - Exacerbate the effects of a system failure

QUALITY

FTA Construction Rules

Include Human Error in FTs

- Human error should be considered in FT model when appropriate
 - When the probability could make a difference
 - When the design needs to be modified
- Key rule – anything left out of the FT causes the results to be understated
- A poor HSI design can force the operator to commit errors
 - Mode confusion (e.g., Predator mishap)
 - Display confusion
 - Too many screens, modes and/or functions
 - GUI Widget confusion
 - Designing the system to complement the human operator

FTA Construction Rules

Human Reliability is Complex

- Finding human error failure data is difficult
- Rates could theoretically vary based on many factors
 - System type
 - Design
 - Human skills
 - Repetitiveness
- In general, studies show:
 - $P = 10^{-3}$ for general error
 - $P = 10^{-4}$ to 10^{-6} if special designs and checks are performed

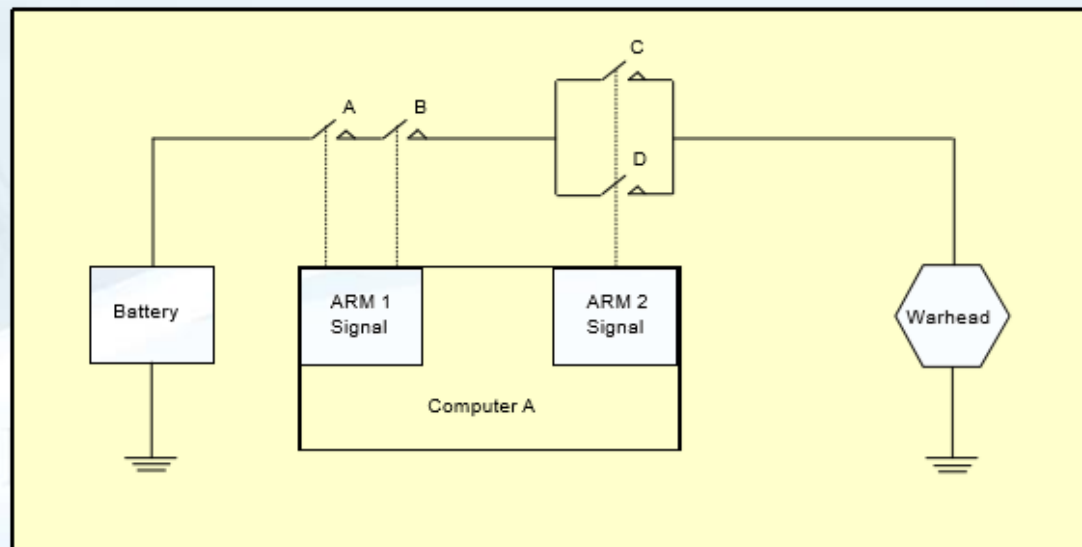
QUALITY

FTA EXAMPLE

QUALITY

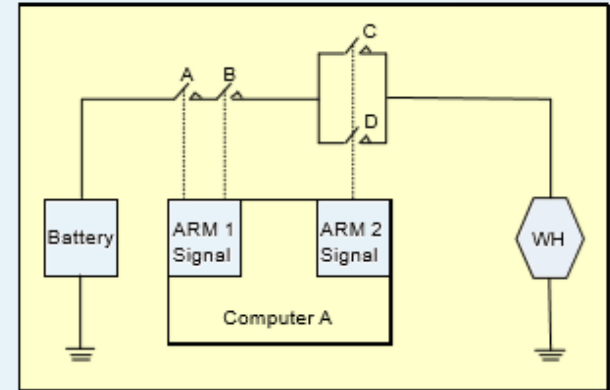
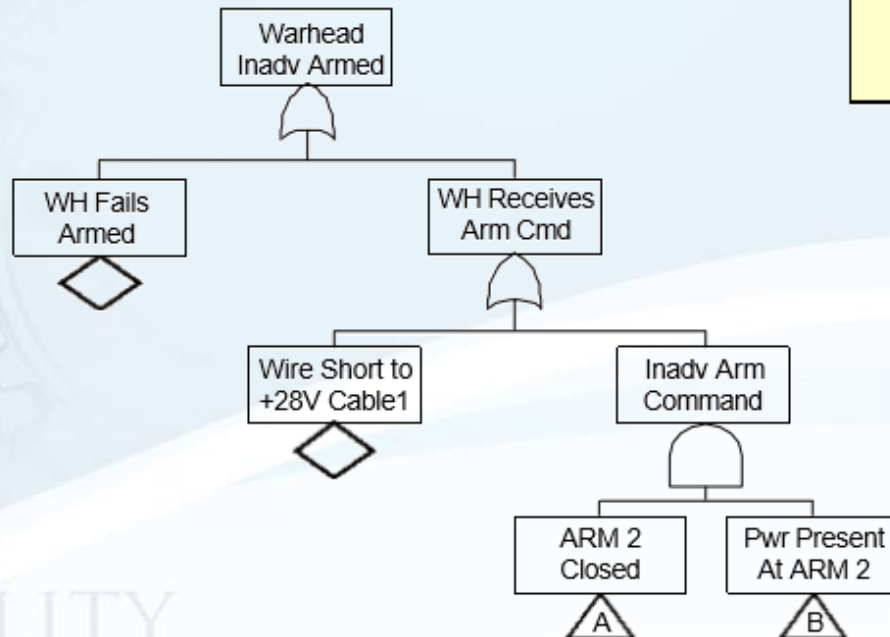
FTA Example

- 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.

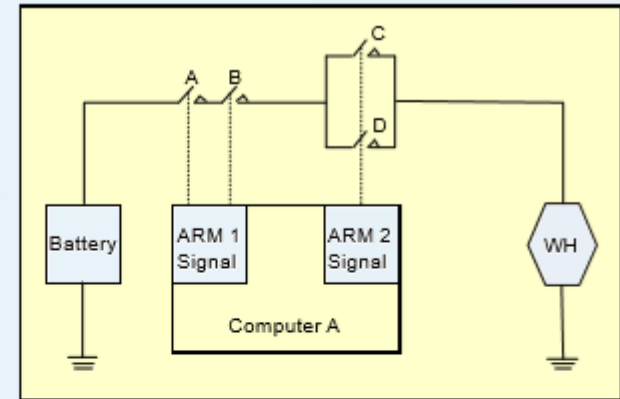
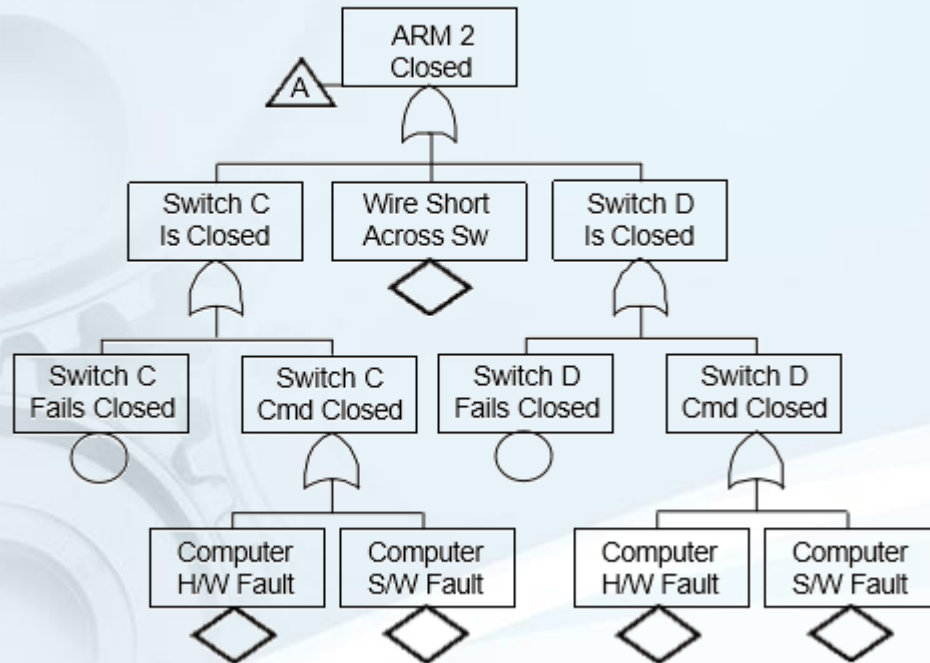


FTA Example

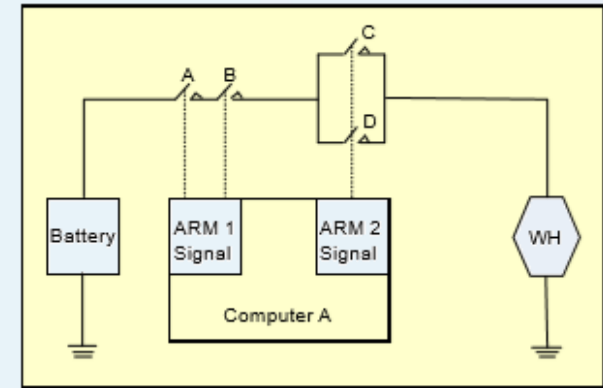
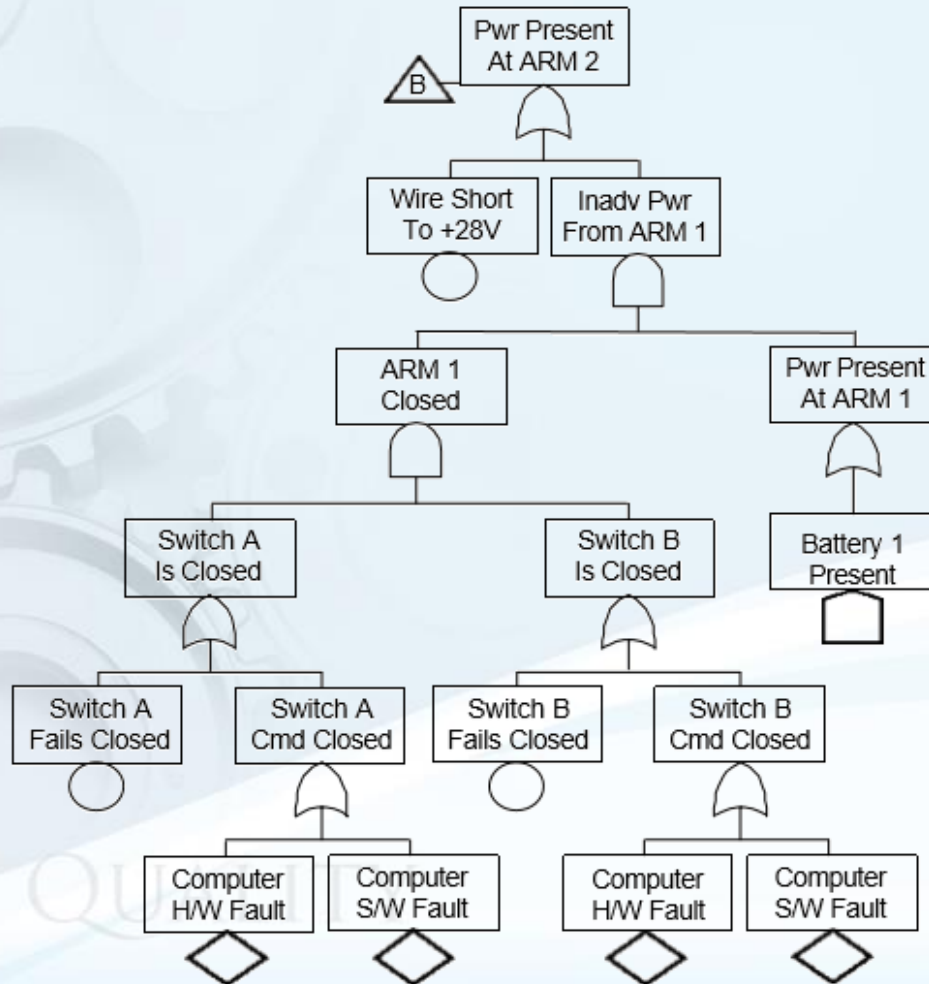
Method 1 – Structured
(Using Functional Approach)



FTA Example



FTA Example



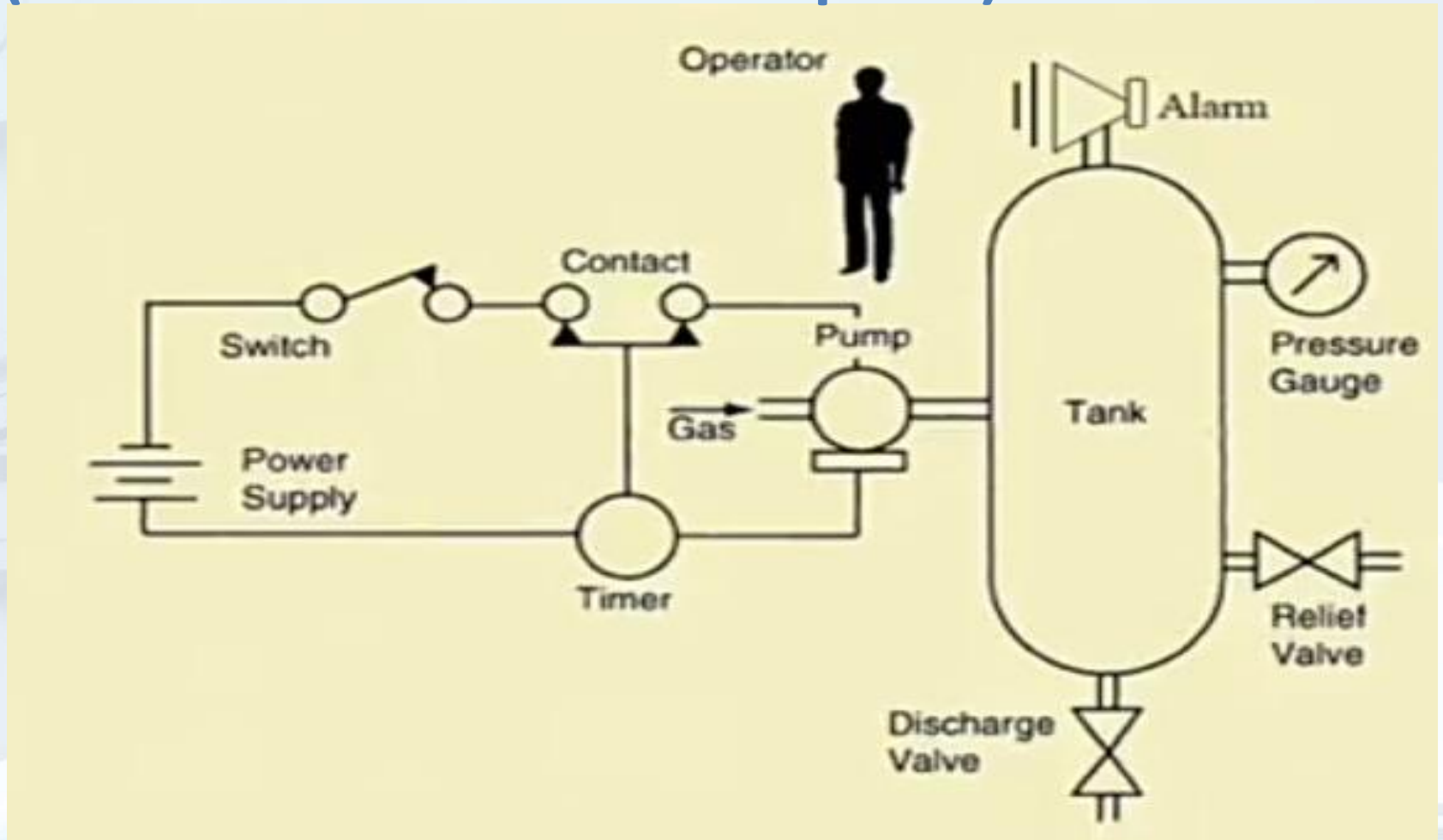
Breakout Exercise 2

Create a the Fault Tree

QUALITY



Pressure tank system (Undesired Event - Tank Rupture)



Breakout Exercise 2: 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 2: 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

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

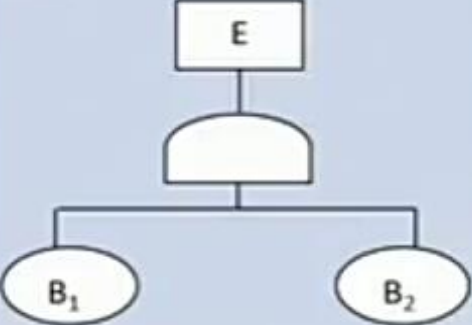
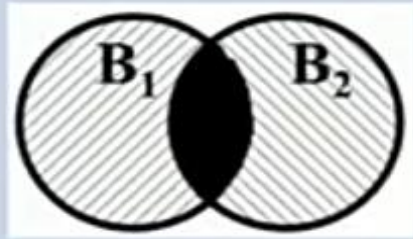
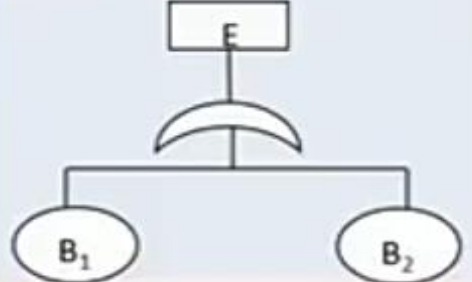
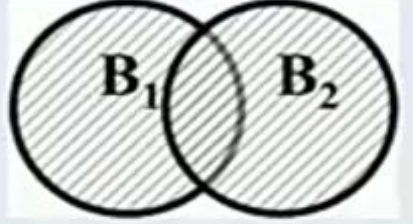
Generate FT results and interpret the findings

QUALITY

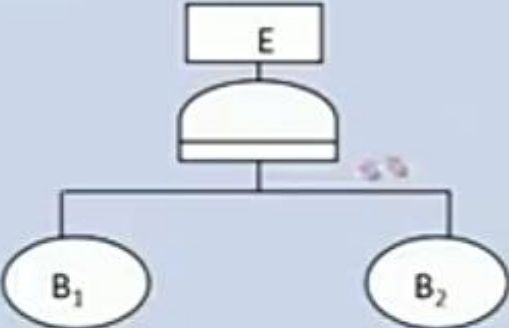
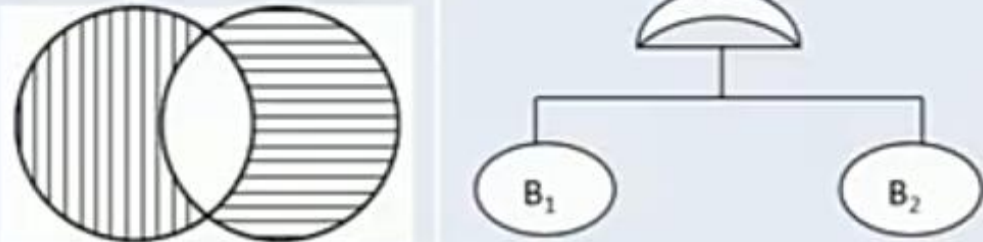
Fault Tree Quantification

- 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.

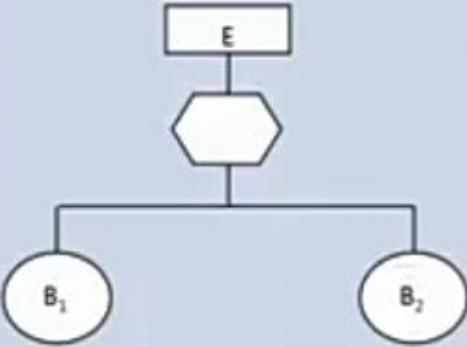
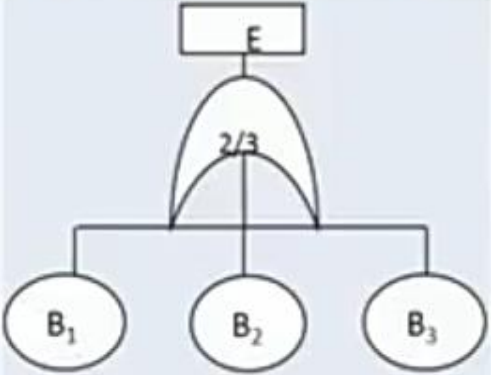
Gate by Gate Method

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 = 2×10^{-3}
- Primary timer failure = 4×10^{-3}
- Primary switch failure = 2×10^{-4}
- Primary operator failure = 3×10^{-4}
- Primary alarm failure = 3×10^{-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.

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.

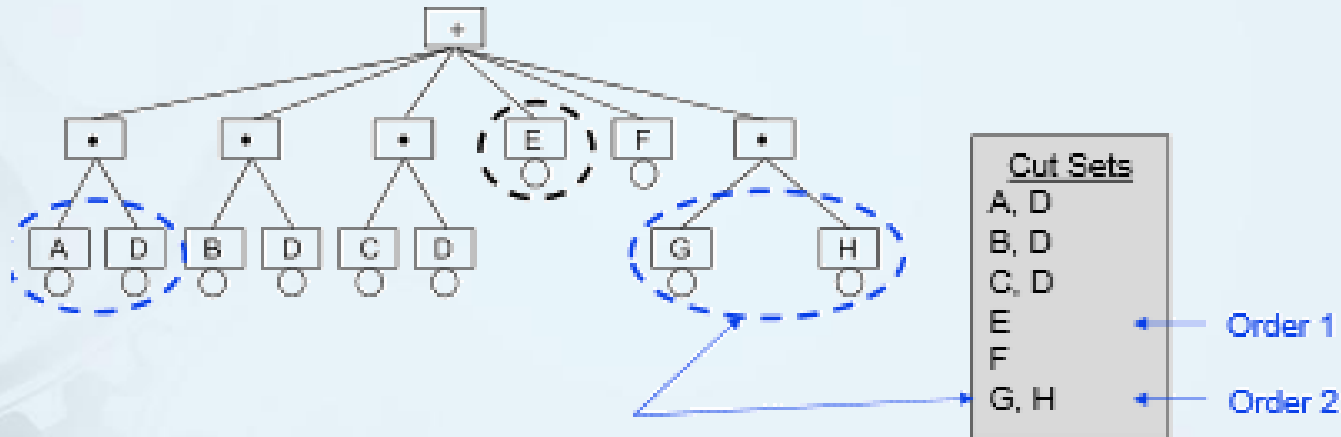
QUALITY

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

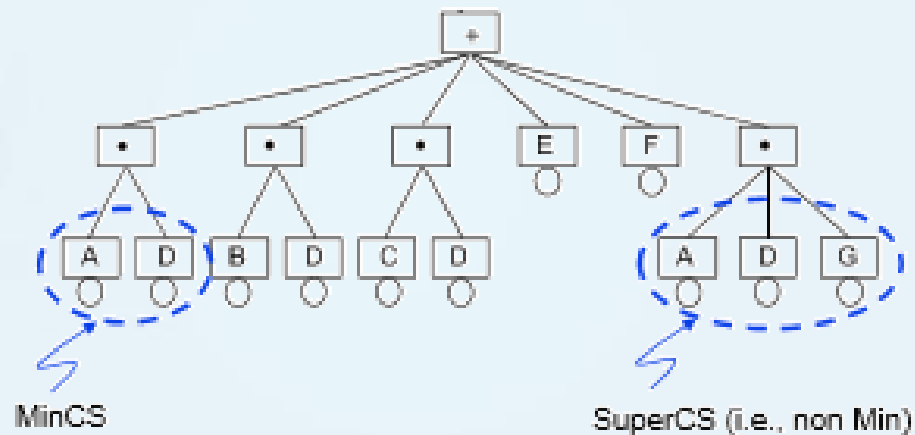


AND gate means that both G & H must occur. Since they go directly to top, they comprise a CS, denoted by {G, H}.

Cut Set (CS)

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

Min CS



Min CS

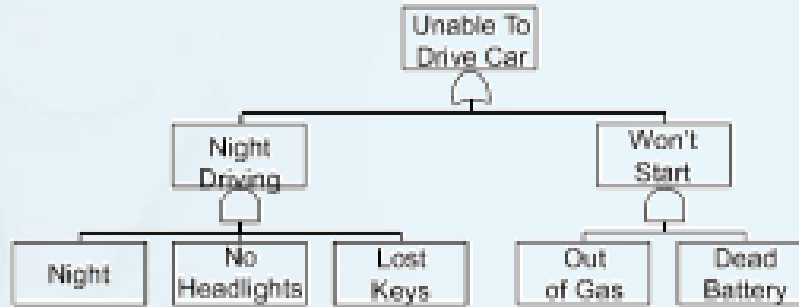
A set of events that contain the minimum number of *necessary* events to cause the Top UE; it cannot be further reduced.

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

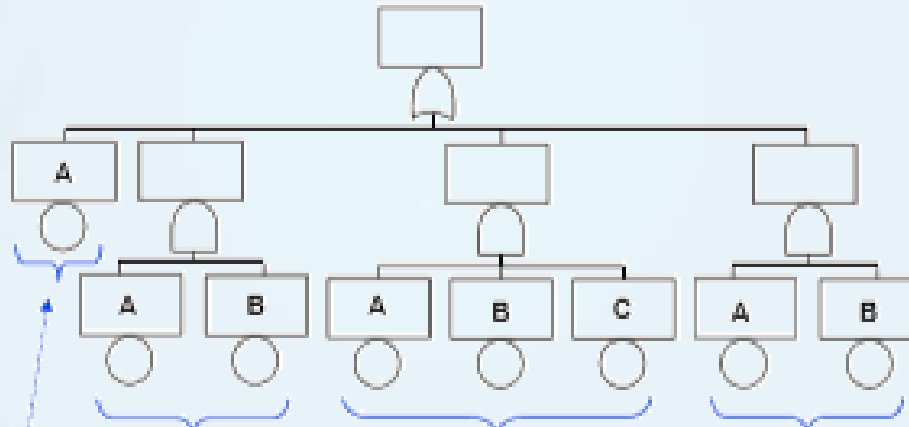
Dead Battery

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 (1×10 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.

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.

Breakout Exercise 4

Development of Risk Mitigation

QUALITY



Breakout Exercise 4: Development of risk mitigation

Handouts

- Develop a Risk mitigation of Previous exercise.

Instructions

- Develop the mitigation for the risk
- Use the flip chart for the exercise
- Be prepared to present your team's to the class; rotate the team spokesperson.

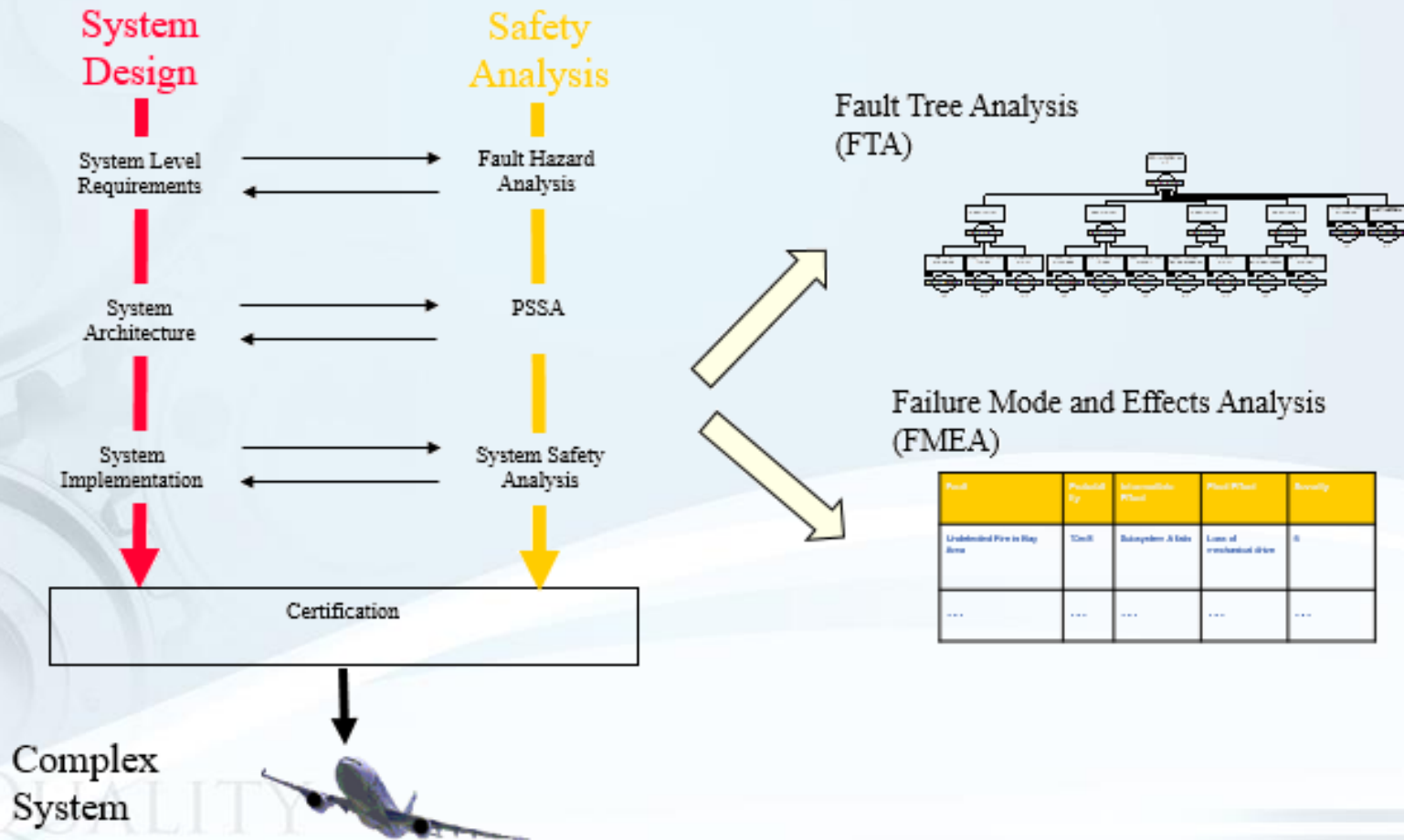
60 Minutes



FTA VS FMEA

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

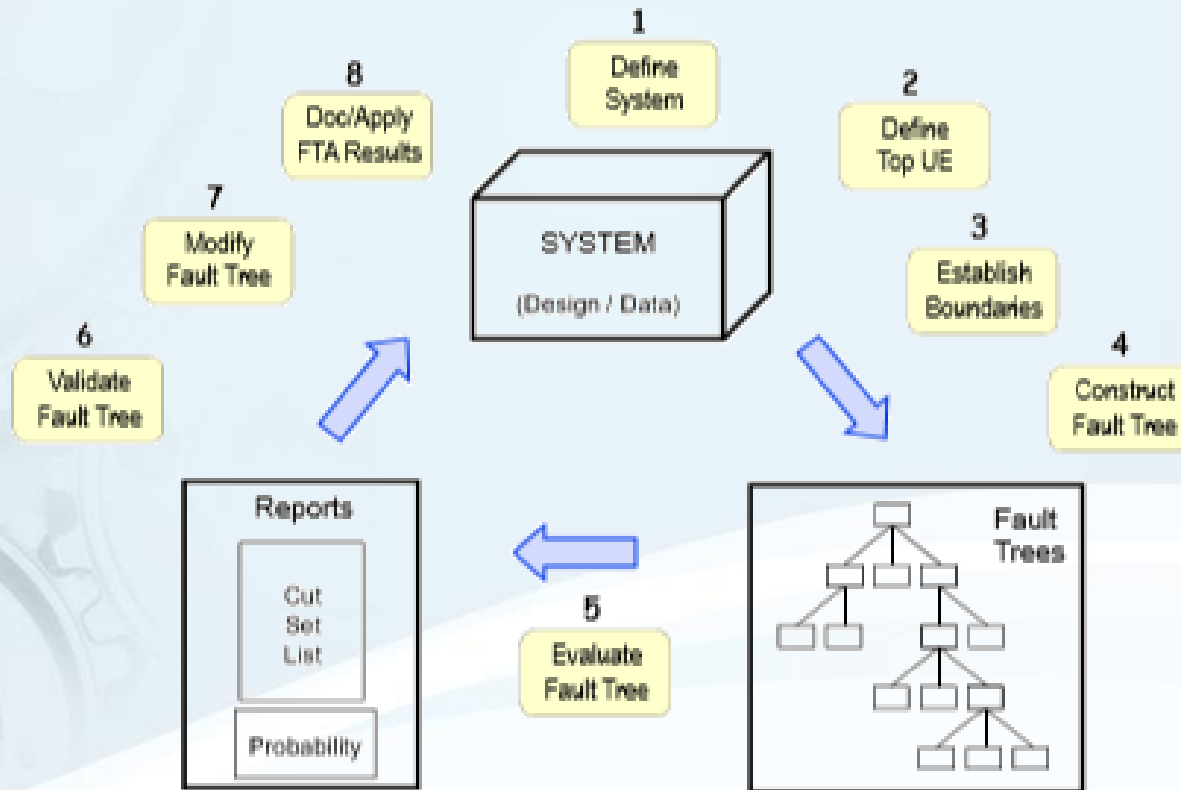
QUALITY

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

