Using Fault Trees to conduct Root Cause Analysis (RCA)

What do we need to do to make sure that 'it' never happens again?

Workbook

Presented by

Dr Chris Jackson

Reliability engineer

acuitas

reliability through insight

Who is your teacher? … Dr Chris Jackson

Dr Jackson holds a Ph.D. and MSc in Reliability Engineering from the University of Maryland's Center of Risk and Reliability. He also holds a BE in Mechanical Engineering, a Masters of Military Science from the Australian National University (ANU) and has substantial experience in the field of leadership, management, risk, reliability and maintainability. He is a Certified Reliability Engineer (CRE) through the American Society of Quality (ASQ) and a Chartered Professional Engineer (CPEng) through Engineers Australia.

Dr Jackson is the cofounder of www.is4.com online learning, was the inaugural director of the University of California, Los Angeles (UCLA's) Center of Reliability and Resilience Engineering and the founder of its Center for the Safety and Reliability of Autonomous Systems (SARAS). He has had an extensive career as a Senior Reliability Engineer in the Australian Defence Force, and is the Director of Acuitas Reliability Pty Ltd.

He has supported many complex and material systems develop their reliability performance and assisted in providing reliability management frameworks that support business outcomes (that is, make more money). He has been a reliability management consultant to many companies across industries ranging from medical devices to small satellites. He has also led initiatives in complementary fields such as systems engineering and performance-based management frameworks (PBMFs). He is the author of two reliability engineering books, co-author of another, and has authored several journal articles and conference papers.

Table of Contents

Introduction

Root Cause Analysis (RCA) is setup as a way to help us answer the question …

… WHY things fail?

An effective RCA goes beyond this. It answers the question …

… what CAN WE DO to stop things failing?

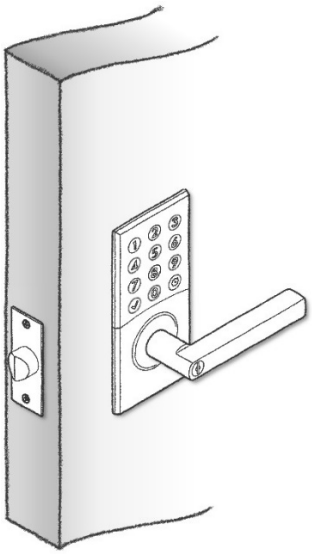
The answers to each question can be very different, depending on perspective. For example, we can attribute blame for failure to someone or some group. But that doesn't help us stop things failing in the future. And that is because a Root Cause is a very specific thing. Most 'textbooks' and references will say something like …

… a ROOT CAUSE is the initiating cause of FAILURE …

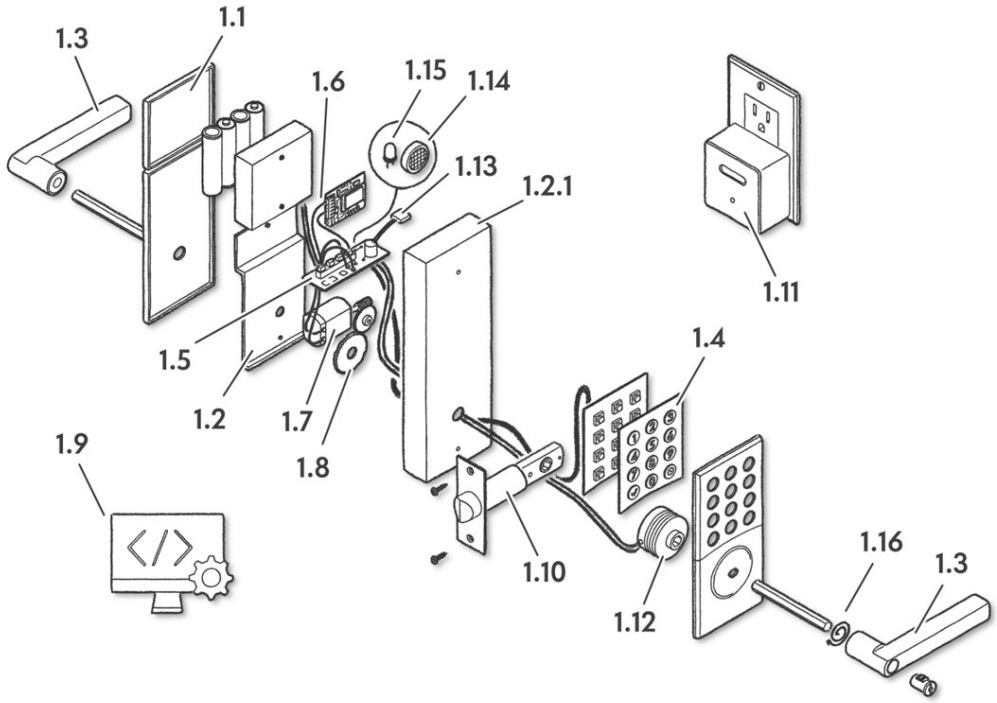
But this is not helpful. Instead, …

… a ROOT CAUSE is the THING we can change to make sure 'that' failure hardly happens again …

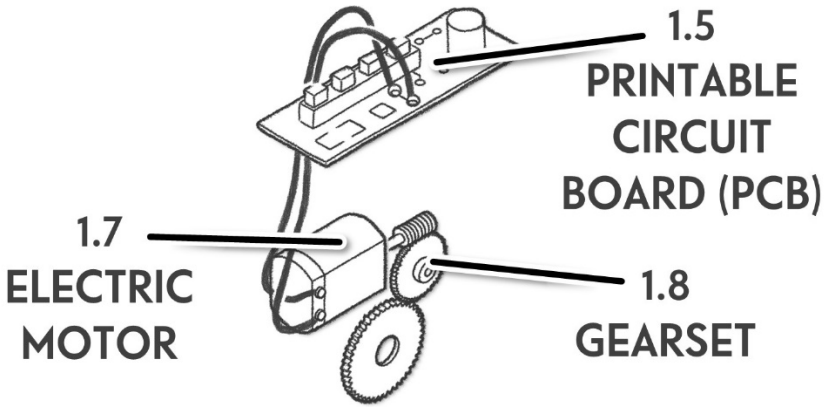
For example, think about a smart lock, that combines traditional mechanical lock technology with 'smart' electronics that allow a new level of functionality for users and customers.



This smart lock is made up of 17 components (16 of them are different as both handles are the same).

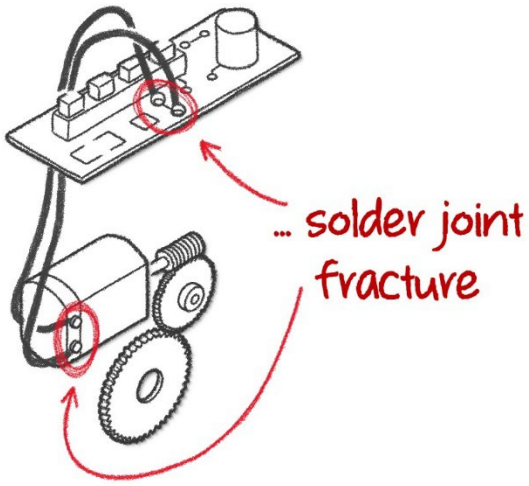


At the 'heart' of the smart lock is the electric motor that is controlled by the Printable Circuit Board (PCB) and the gearset it turns.



Of course, we want to prevent this part of our smart lock from ever failing. This means we could conduct RCA on potential ways it could fail, which is what happens in Failure Modes and Effects Analyses (FMEAs) or Fault Tree Analyses (FTAs) that are conducted early in design to come up with a robust first design that never allow those failures to occur. And we might also need to conduct RCA on failures we observe during early prototype testing to update and change future designs to ensure that we eliminate or reduce the likelihood of that failure occurring again.

So let's say that we have either envisaged a potential failure we want to avoid, or encountered a failure we need to prevent in future designs.

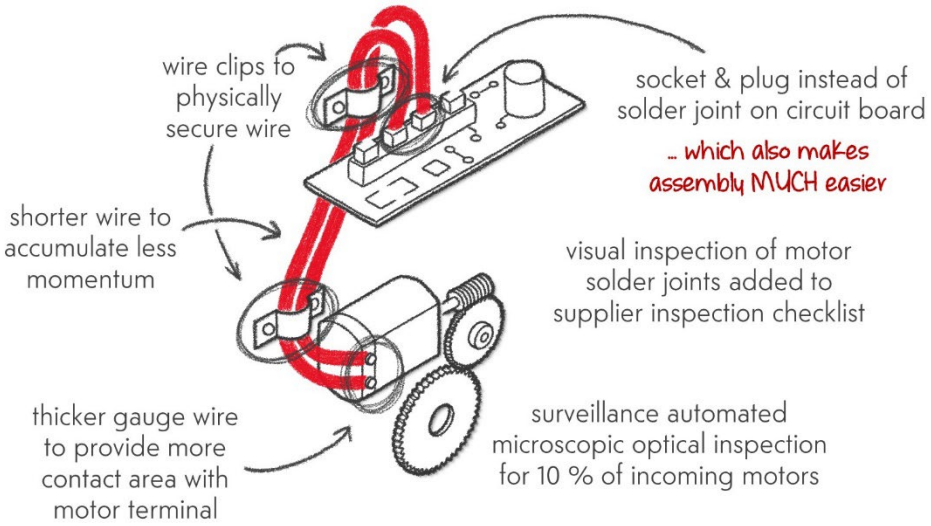


Is 'solder joint fracture' a Root Cause of failure? NO. Because we can't change this. This is a symptom of something we need to do different. Simply resoldering these joints won't prevent whatever issue exists in our design from causing this failure again in the future.

Are the 'door slams' that might have created the vibration that caused these solder joints to fracture the Root Cause of failure? NO. We can't influence the behaviours of our users or customers. We need to design products that work in the ways our customers and users want to use them. So if our customers want to slam doors, then we need to deal with that!

What about something more along the lines of the solder joints doing some they weren't designed to do? Or in other words, the 'design is incapable to deal with shock loads at cable connections?' Now we are getting somewhere. We can 'control' the design.

The ideas we come up with to prevent failures from occurring again are called Corrective Actions (CAs). And some examples for the fractured solder joints are illustrated below.



… but this might be TOO SIMPLE

We often don't have the ability to 'see' what goes wrong. For example, we might launch our smart lock, and then start receiving more warranty claims then we thought we should have. We might not get the physical locks back, or even a description of what went wrong.

But that is where structured, well facilitated RCA can be incredibly helpful. Many engineers believe that RCA is all about finding the 'provable,' single, 'true' Root Cause of failure. This doesn't happen very often. Instead, we need to be happy to find a suite of likely potential root causes, and then prioritize the CAs we implement for each.

The earlier you do this (through things like FMEAs and early production FTAs), the more flexibility we have to chose CAs that are easier and cheaper to implement.

Lots of different approaches and techniques have been developed, along with their acronyms (5Ys, 5WHYs, 5WHYs & 3 tiers, Statistical Process Control or SPC, 8Ds, 4/5/6/8Ms, 5W1H, DMAI, Apollos, Pareto Analysis, Ishikawa Diagrams, and plenty more).

But the most important element of RCA is the quality and reliability mindset which we will illustrate with the light-bulb icon below.

The 'language' part of the light bulb refers to the common set of term and concepts we all share, so that when one person is brainstorming as part of RCA, they are not confusing or counteracting the thoughts and words of others. The 'methods' part of the light bulb refers to the structured approach we must use to ensure our brainstorming efforts quickly and efficiently focus on the most likely Root Causes we need to address. And finally, we must always understand the 'value' of what we are trying to achieve to understand what is worth doing, and what is not. And most importantly, being able to understand how much 'value' our CAs will generate.

… so how do we DO THIS RIGHT (… not WRONG)?

There are some basic things we need to do to get a good, valuable outcome of RCA.

1. state the PROBLEM (and the decision you are able to make)

This needs to be set in reality. For example, if RCA is left too late … don't do it! This cheapens and diminishes RCA in your organization and associates it with being a waste of time. So not understanding the problem, or decision you are trying to inform dooms any RCA to mediocrity or irrelevance.

2. work through LAYERs (and not rushing to your FAVOURITE solution)

Most failures (or undesirable events) are the result of many possible chains of events. These chains can branch (or root) out, and quickly following one chain of events will result in a small number of potential root causes that might not actually represent the true reason behind something failing. Instead, we need to go down 'one layer at a time' and brainstorm all possible preceding events. Each new event we brainstorm then becomes the start of a new chain of events we need to pursue. While this sounds like more work, the result is lots more potential Root Causes (and CAs) for us to choose from, including many that will be faster, cheaper and more effective.

3. don't be a LAWYER (who wants to find a SINGLE ROOT CAUSE to allocate BLAME)

So what if the 'electric motor' is causing failure? What if the easiest fix involves changing the design of the gearset? This happens a lot, where one component or subsystem isn't behaving the way we like, and the easiest, fastest, cheapest and most efficient CA involves modifying the design of another component or subsystem. The same thing applies to suppliers. Yes, it might feel 'just' to apportion blame to a supplier whose component is not behaving. And the money they spend on remedying the issue will eventually be paid for (by you). There will of course be some times where suppliers need to be accountable for mistakes they might make, but there is a reason why lawyers don't make great designers or manufacturers.

4. find LOT's OF PROBABLE ROOT CAUSES (so we can pick the cheapest and easiest ones to fix).

5. you aren't done until you FIX the ROOT CAUSES you find and select

RCA is not about admiring problems. It is about finding CAs and then managing them until they are validated. In some cases, the probable Root Causes we identify CAs for might not end up being the 'true' Root Causes. So we then need to move onto the next most probable Root Causes and identify CAs for them. We keep going until the failure has been addressed and prevented.

What is a root cause?

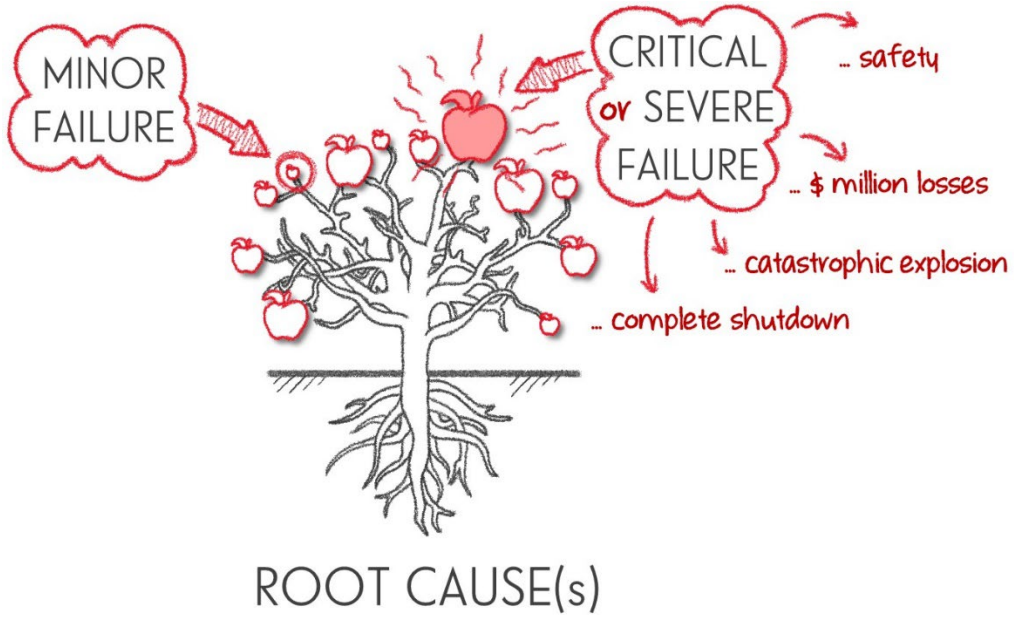
From the previous chapter, we know that most 'textbooks' and references will say something like ...

... a ROOT CAUSE is the initiating cause of FAILURE ...

But this is not helpful. Instead, ...

... a ROOT CAUSE is the THING we can change to make sure 'that' failure hardly happens again ...

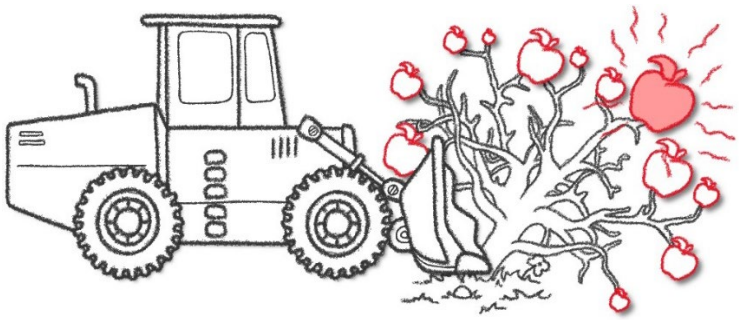
A Root Cause is something that you own, or something you can do something about. The term 'root' comes from a popular we visualize how 'bad things happen.' And that way we visualize comes from a tree.



Most trees have more than one root. But the term 'Root Cause' is a singular reference that is written as if there is only one, 'true' Root Cause. In reality, there are often lots of different Root Causes that combine to result in the undesirable event we are analyzing. And by identifying lots of potential Root Causes, we might be able to choose the Corrective Actions (CAs) that are the cheapest, easiest and fastest to implement.

… then there are IMMEDIATE ACTIONs

Whenever an undesirable event (like failure) occurs, we usually need to do something quickly to contain its consequences. This is like removing the 'visible' part of the tree above the ground.



The problem is that if we do not address the Root Cause(s) of failure, the tree (and the fruit that represent failures) will simply come straight back. This doesn't mean that we shouldn't do something quickly to contain those consequences. Its just that these will be a short-term solution. And we call these short-term solutions 'immediate actions.' Examples include …

… CORRECTIVE MAINTENANCE (CM) or REPAIR where we address the physical and functional issues that result in system or equipment failure …

… ISOLATION where the failed system or equipment is removed or separated from the rest of the assets to ensure that it's failed state doesn't affect operations …

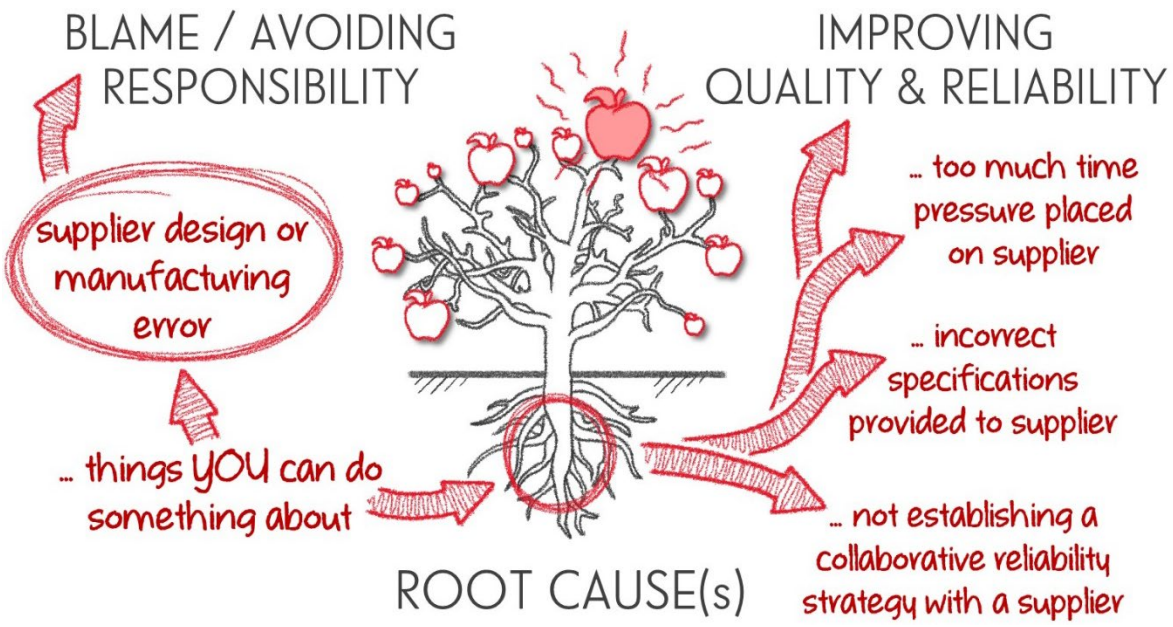
… CONTAINMENT where mechanisms or structures are placed to ensure the consequence of failure doesn't cause further damage or harm …

… RECALL where a manufacturer or supplier takes the systems or equipment back from customers in order to implement actions that will mitigate further failures …

There are lots of others that are specific to each scenario, and are usually necessary steps before more detailed RCA.

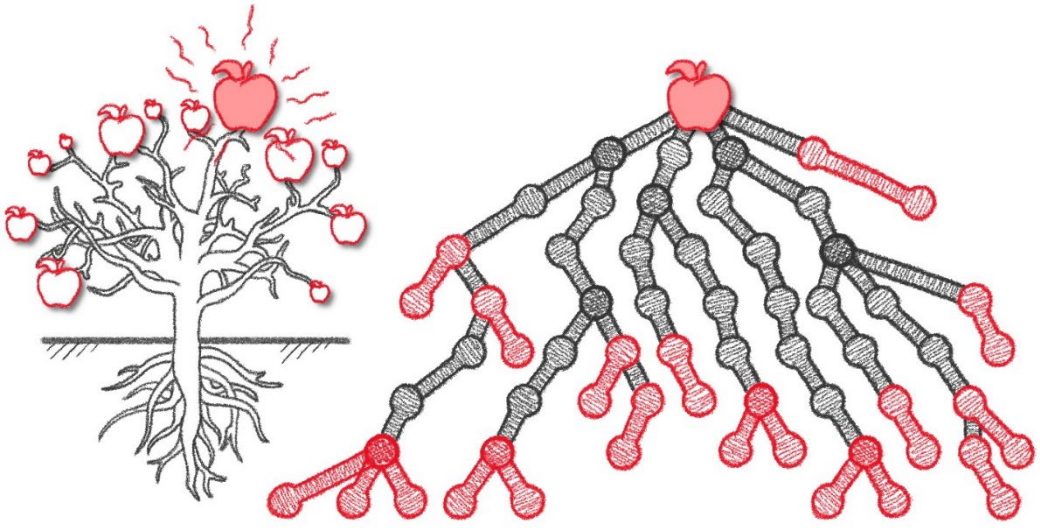
… ROOT CAUSES are all about what 'WE' can do …

Truly wanting to find Root Causes of failure usually means we need to be humble about what went wrong.

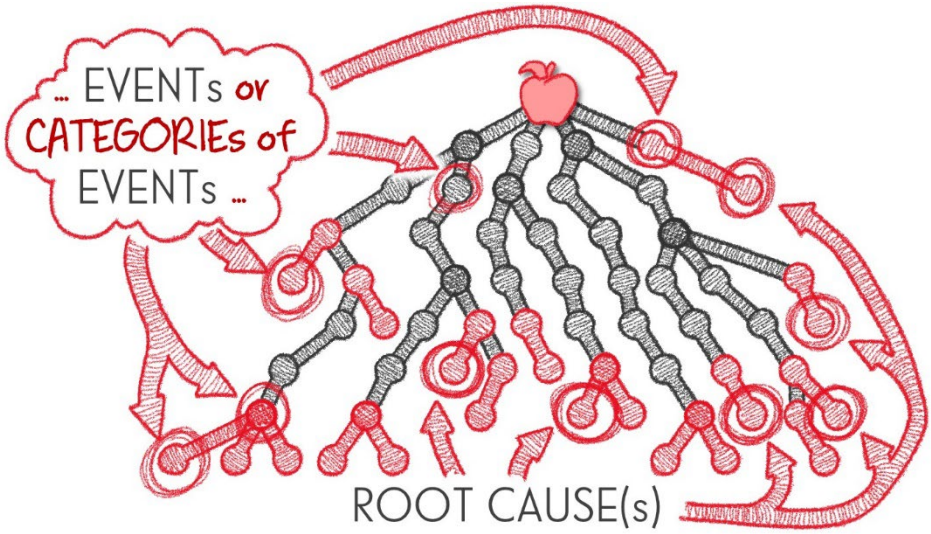


… let's talk about LAYERs

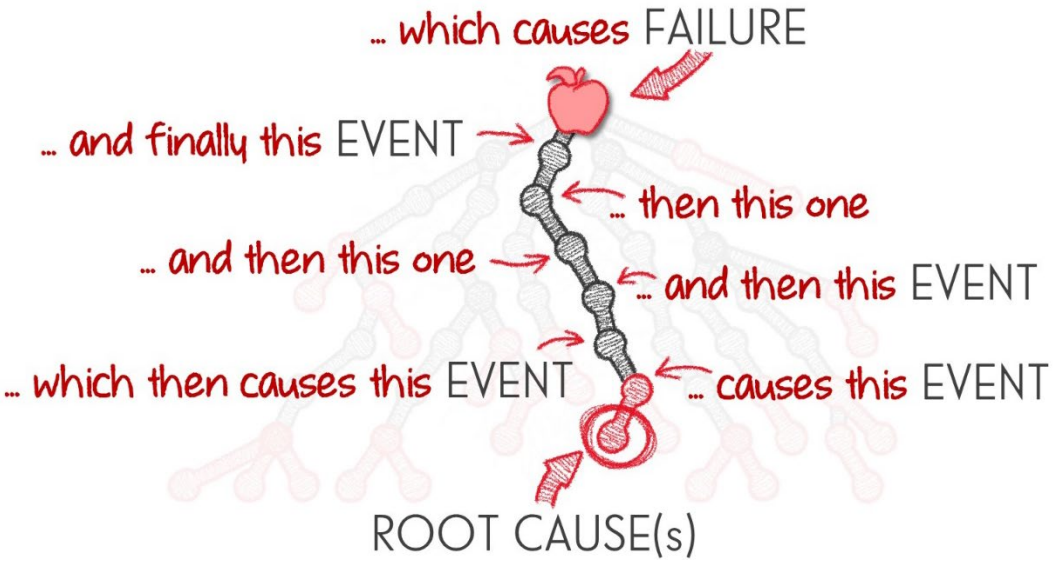
While the tree of failure is a useful visual representation of how failures relate to Root Causes, the layered root system (below, on the right) is a better way of representing how we actually uncover potential Root Causes of failure.



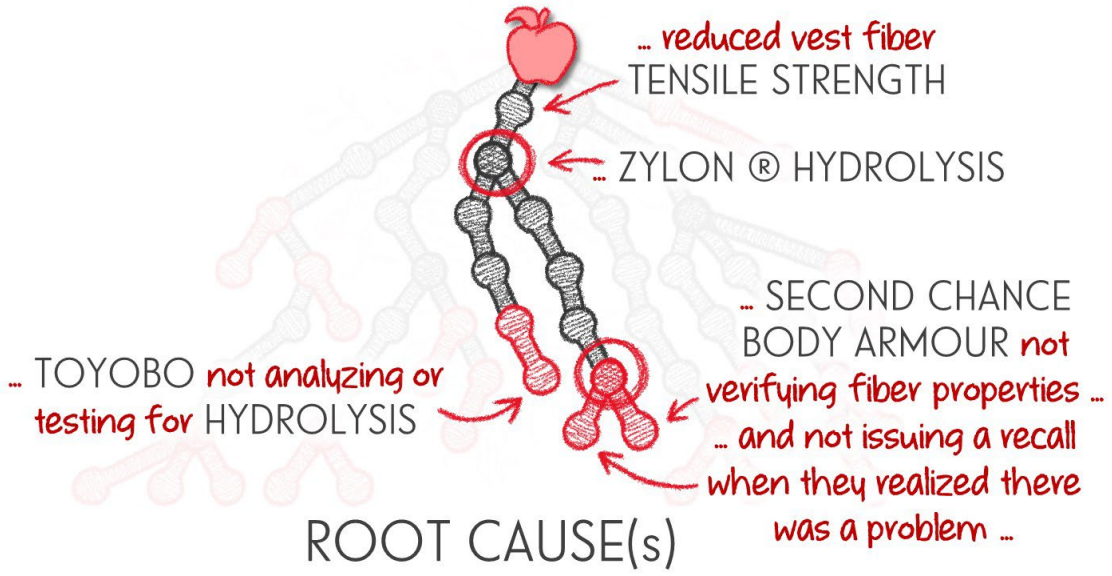
The layered root system on the right contains multiple branches of or categories of events.



Each branch is what we call a 'Causal Chain' which is a sequence of cascading events where one event becomes the trigger for the next event. The first event in this Causal Chain is the Root Cause, while failure is the last or ultimate event.

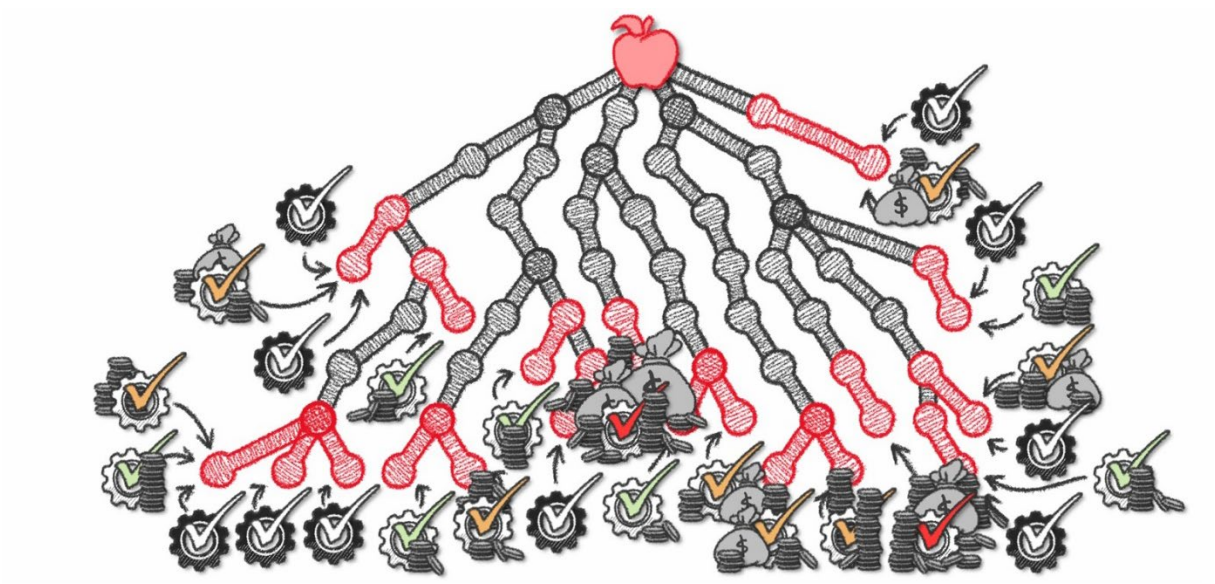


And Causal Chains can interact.



... LAYERs are the way we find CORRECTIVE ACTIONs

We want to find as many potential Root Causes as possible, to find as many potential Corrective Actions (CAs) as possible. This gives us as flexibility to find the cheapest, fastest, easiest and most effective CAs so resolve the issues. Each 'cog with check or tick mark' below within a gear represents a potential CA. The amount of money (and colour of the tick or check mark) represent the relative ease and costs of each CA.



The earlier we can uncover CAs, the cheaper, faster, easier and more effective they are. This is where activities like Failure Modes and Effects Analyses (FMEAs), and Highly Accelerated Life Testing (HALT) that are conducted early in the production life cycle become incredibly valuable.

Example 'early' CAs include ...

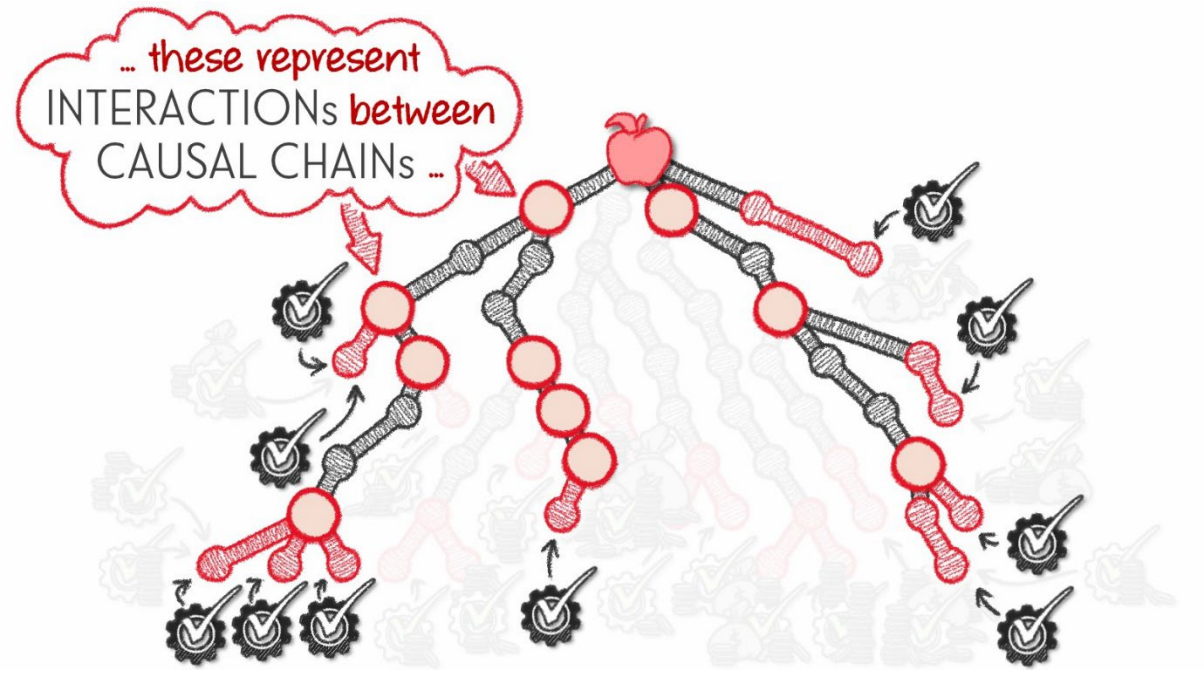
... mounting the PCB on the left side of the engine bay (away from the HOT EXHAUST MANIFOLD) ...

... using an AUTOMATED MIXER to ensure the correct amount of vulcanizing agent is mixed ...

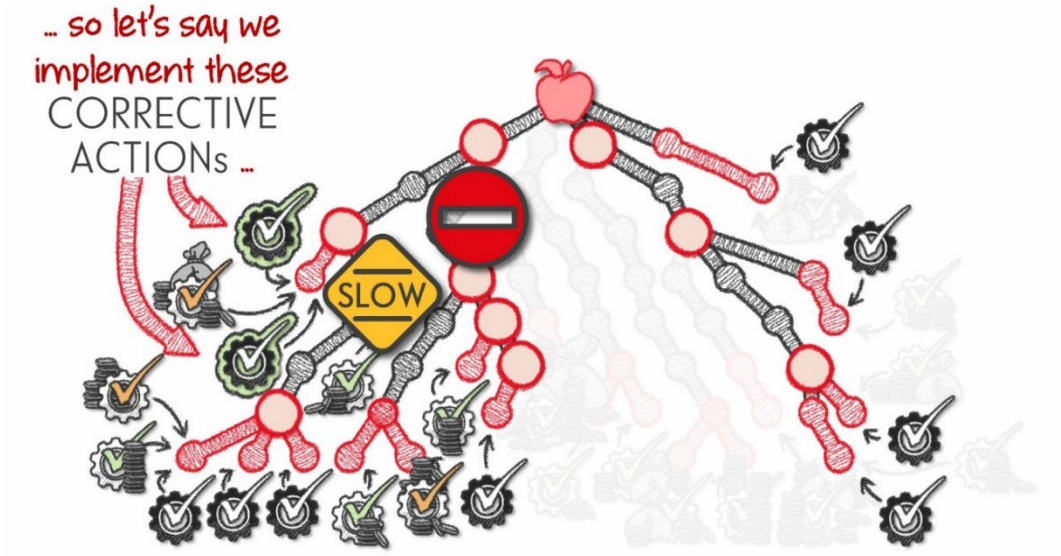
... colour coding wires in the WIRE HARNESS to make it less likely for wiring mistakes to occur during installation and repair ...

... conducting FINITE ELEMENT ANALYSIS (FEA) on the door handle to ensure all stresses are below the ENDURANCE LIMIT of the steel alloy (so FATIGUE can't occur) ...

The interactions between the Causal Chains also help us identify opportunities to again, find the cheapest, fastest, easiest and most effective CAs. The simple 'black' CAs in the illustration below represent the cheapest, fastest, easiest and most effective CAs that can be implemented.

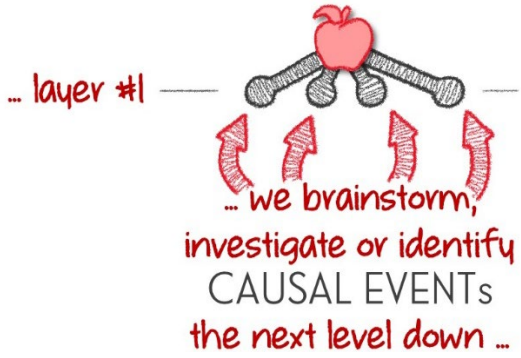


The interactions above allow us to use those cheap, fast, easiest and most effective CAs to potentially negate other linked Causal Chains by either stopping them entirely or slowing them down.



... and back to LAYERS

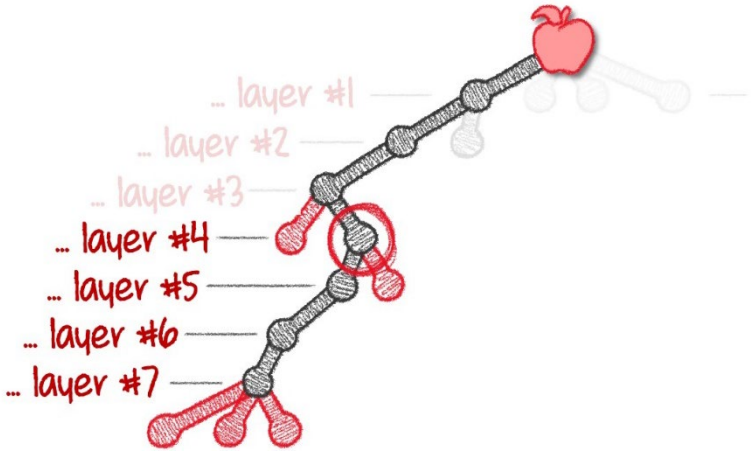
To find as many Potential Root Causes as possible, we need to brainstorm these Causal Chains one layer at a time. For example, if we start Root Cause Analysis of our failure event we start by finding 'layer #1.'



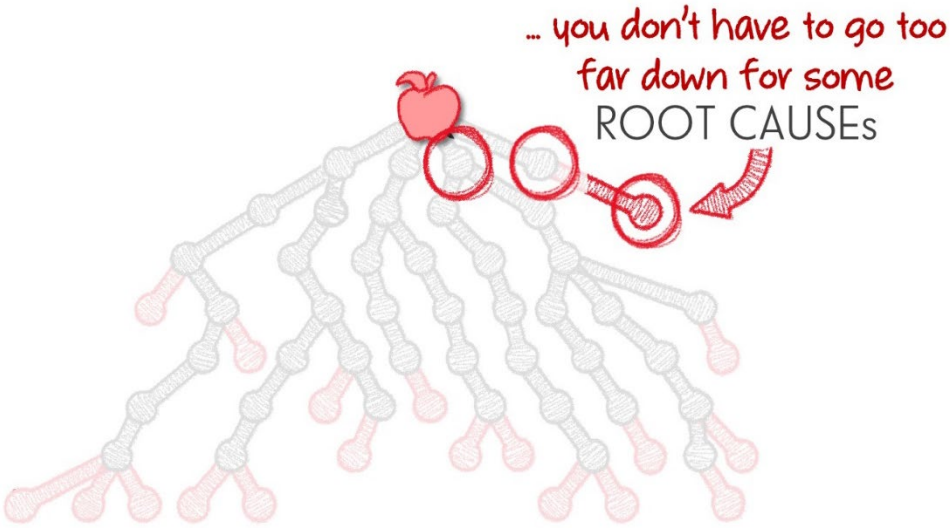
Once we are satisfied we have found all potential events in 'layer #1' that could immediately cause the failure event, we then focus on just one event and then focus our brainstorming on that event to move onto 'layer #2.'

We then focus on one of these events in 'layer # 2' and keep working our way down until we are satisfied that one of the events we brainstorm is a Root Cause that we can actually do something about.

We then move the next adjacent event, and keep going, with our brainstorming stopping and moving on once we arrive at Root Causes.



You can see that we find Root Causes can be found at different layers.

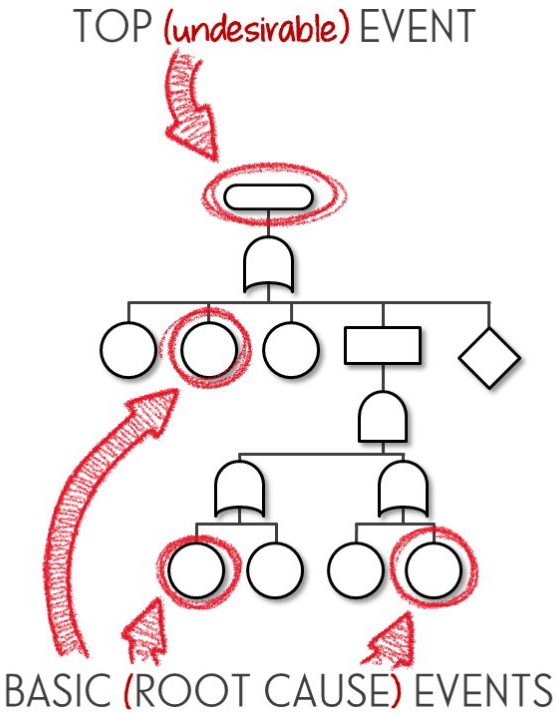


The most important element of the concept of layers when we brainstorm potential Root Causes is that we don't 'rush straight down' to the most obvious or our 'favourite' potential Root Cause. If we stop and think (and allow our team to brainstorm), we can come up with lots of different reasons failure occurs, giving us lots of flexibility to find those easy, fast, cheap and effective CAs.

These are what we call the VITAL FEW.

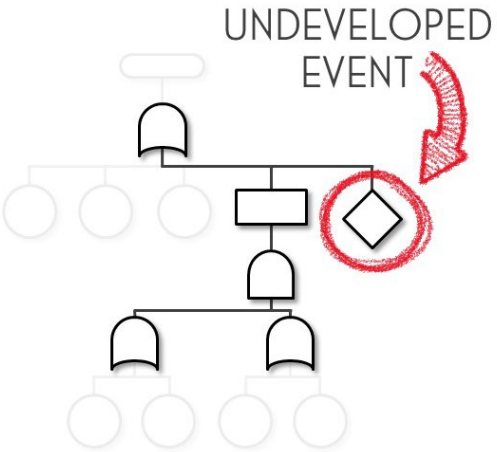
Visualizing what (we think) happened

Fault Trees are similarly powerful and sophisticated as Ishikawa Diagrams as they allow lots of causal chains to be developed, with as many layers as desired. Their basic structure is as follows ...

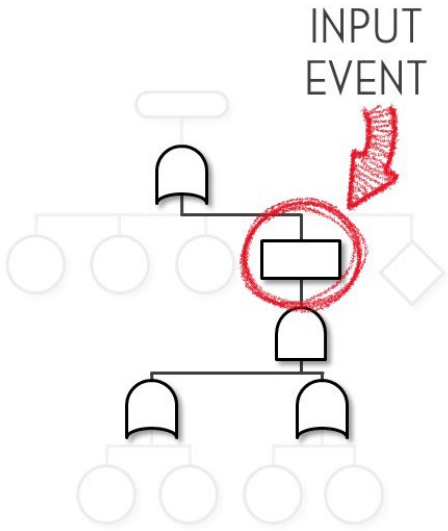


So the diagram above represents potential Root Cause with circular or round shapes that are called 'basic events.' The structure and layout of the Fault Tree shows us what Basic Events can result in the Top or 'undesirable' event which represents the failure event which triggered the Root Cause Analysis.

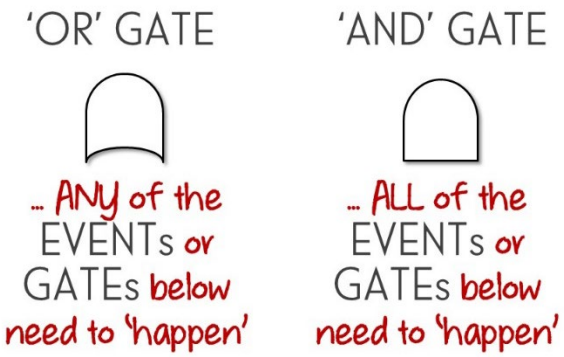
There are some other shapes in the Fault Tree as well. An 'Undeveloped Event' is an event that we know we need to potentially explore further in the future. Perhaps we didn't have the expertise on our team at the time to follow this particular causal chain further. Maybe this 'Undeveloped Event' represents some sort of electronic failure, and our electronic engineer is unavailable that day. Or perhaps it deemed improbable, and we will only explore it further if we get more information that makes it more probable.



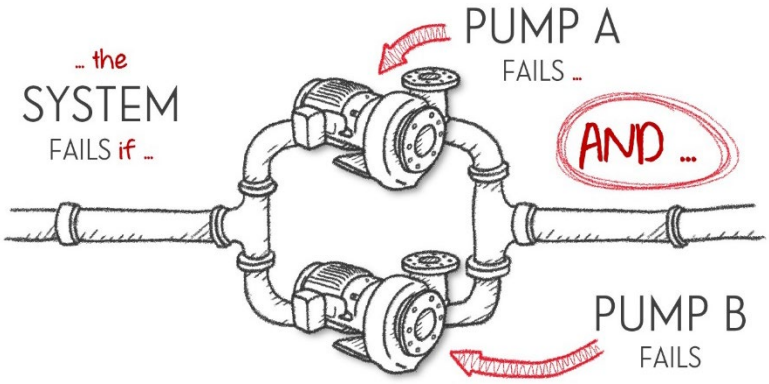
There are also 'Input Events' we represent with rectangles that allow us to label what each part of our Fault Tree represents. For example, an 'Input Event' might be 'Power Supply Failure' because the elements of the Fault Tree beneath it represent different ways that the power supply could fail.



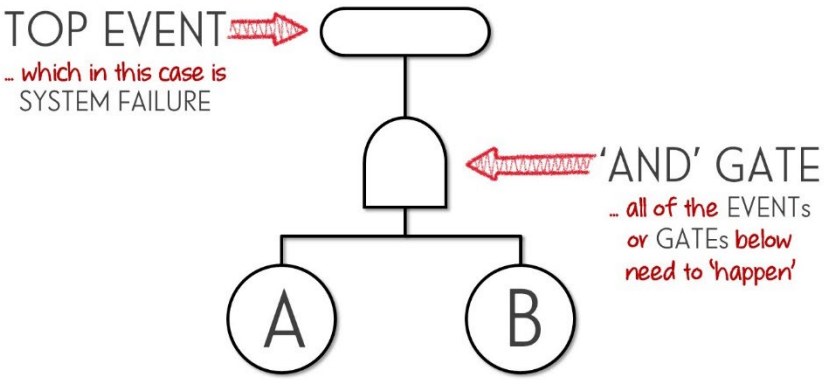
Then there are the following two shapes that are what we call 'Logic Gates.'



To understand how these gates work, let's look at a two pump 'Parallel System' where one pump is redundant.

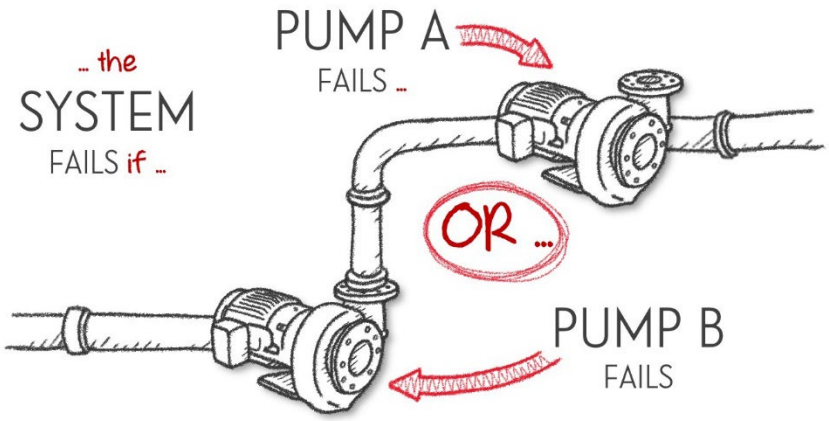


A Fault Tree can model this system using an 'AND' gate ...

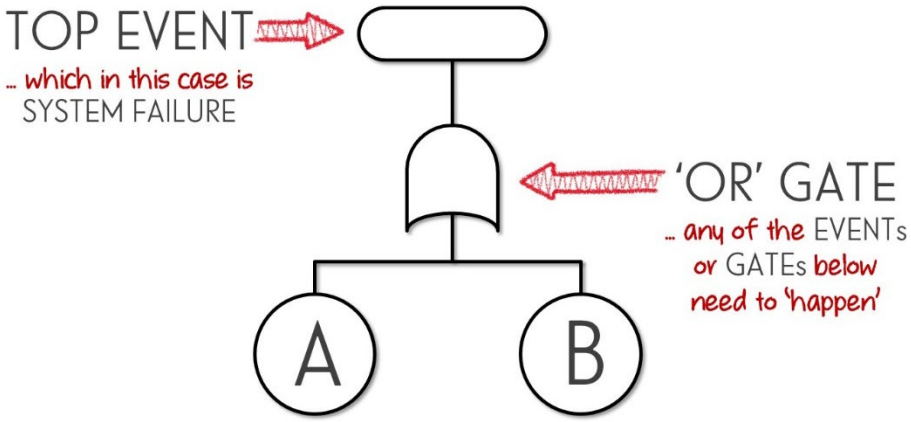


The circles with the letters 'A' and 'B' in them represent 'Basic Events' corresponding with the failures of pumps 'A' and 'B' respectively.

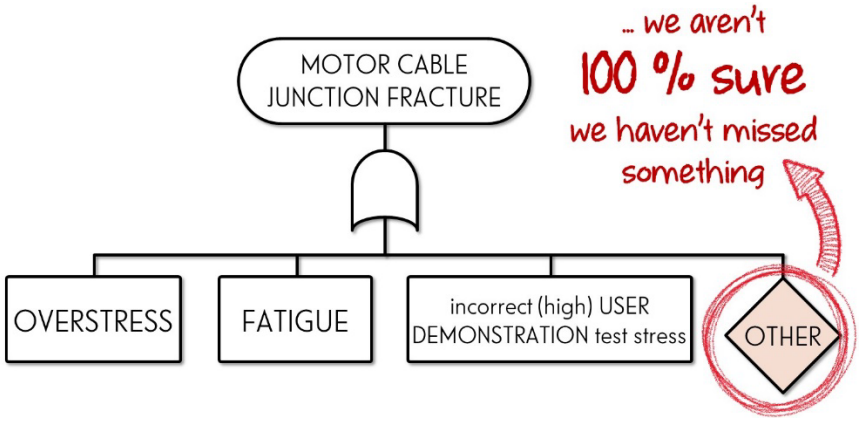
Then there is a 'Series System' that has no redundant pumps.



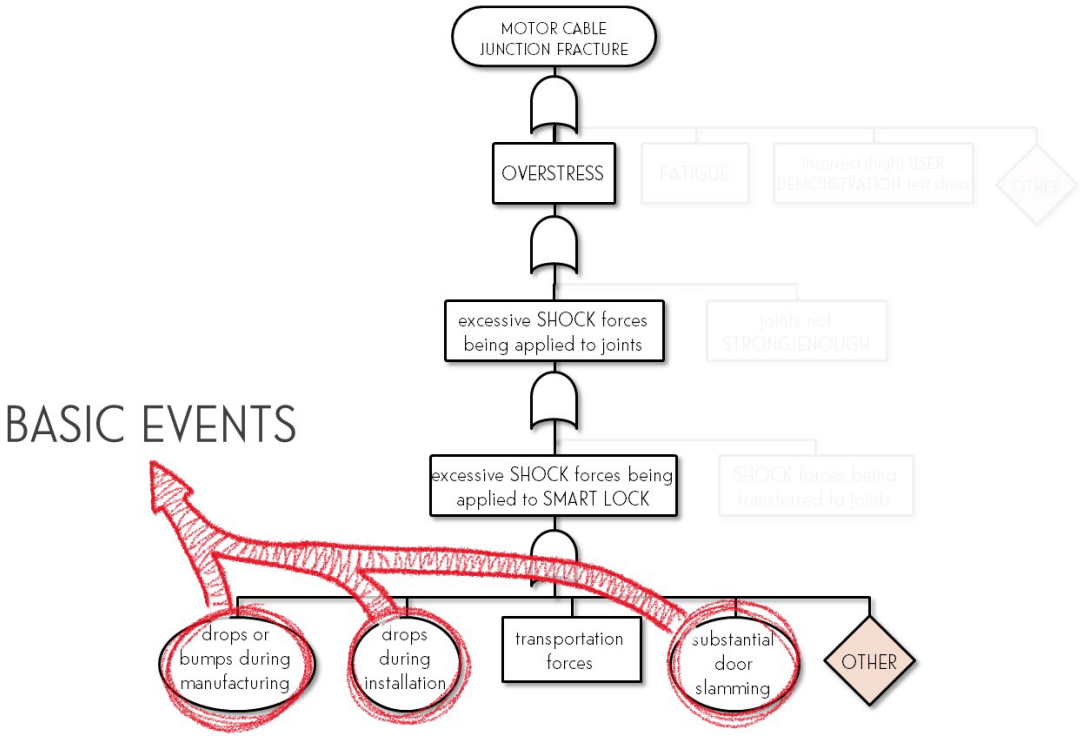
A Fault Tree can model this system using an 'OR' gate ...



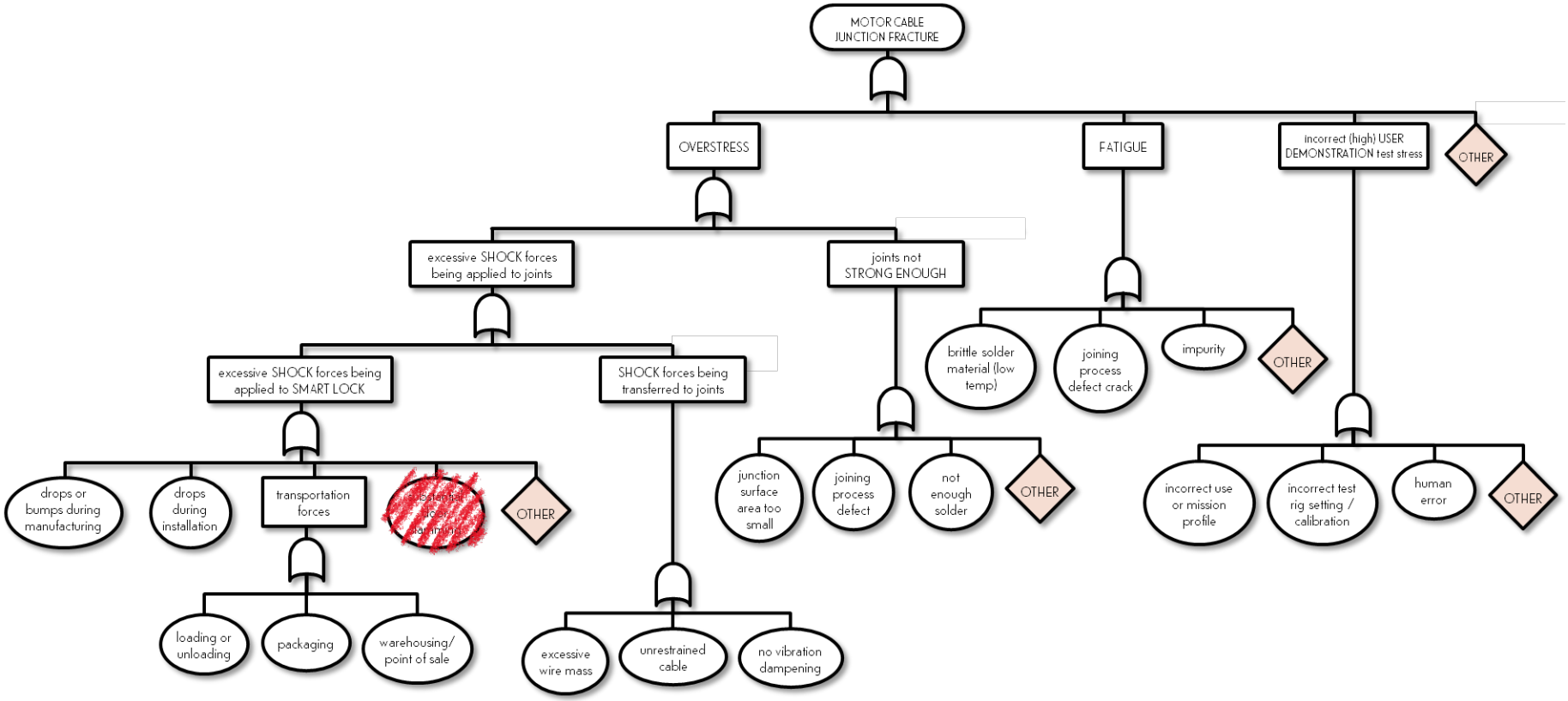
Fault Trees, like Ishikawa Diagrams, allow us to go down one layer at a time. For example, we might brainstorm the following layer #1 'Input Events' and 'Undeveloped Events' as part of Root Cause Analysis for our motor cable junction failure.



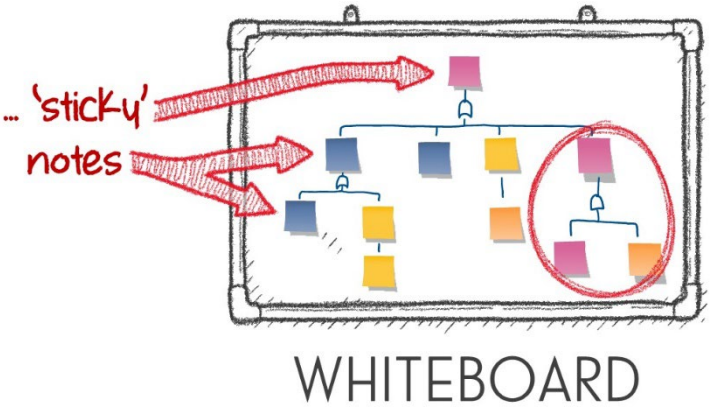
We then focus on one of the 'Input Events,' such as 'Overstress' and then brainstorm layers until we arrive at events that we believe satisfy the definition of being a Root Cause. We then replace the rectangular shapes of 'Input Events' with the round, circular shapes of 'Basic Events' to show that those events now need to become the basis of generating potential Corrective Actions.



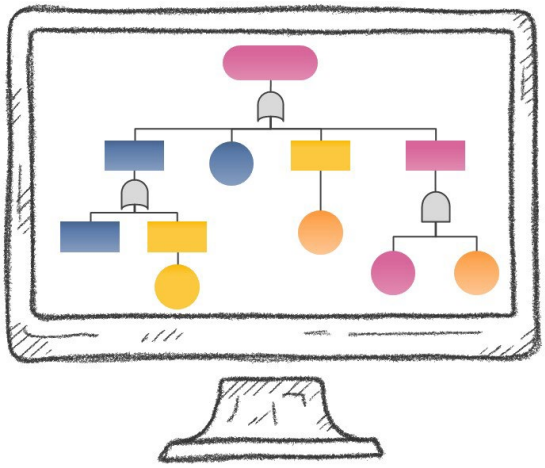
The fully developed Fault Tree is shown on the following page.


MOTOR CABLE JUNCTION FRACTURE
OVERSTRESS
FATIGUE
incorrect (high) USER DEMONSTRATION test stress
OTHER
excessive SHOCK forces being applied to joints
joints not STRONG ENOUGH
excessive SHOCK forces being applied to SMART LOCK
SHOCK forces being transferred to joints
brittle solder material (low temp)
joining process defect crack
impurity
OTHER
drops or bumps during manufacturing
drops during installation
transportation forces
OTHER
junction surface area too small
joining process defect
not enough solder
OTHER
incorrect use or mission profile
incorrect test rig setting / calibration
human error
OTHER
loading or unloading
packaging
warehousing/ point of sale
excessive wire mass
unrestrained cable
no vibration dampening

Fault Trees are also arguably easier to modify and amend during analysis, particularly when the facilitator uses 'sticky notes' and 'whiteboards.'



Further, Fault Trees are easier to use when Root Cause Analysis is conducted virtually using Microsoft ® PowerPoint, Google Slides, Figjam or other online collaboration software.



Ishikawa Diagrams often need proprietary software, especially when the Root Cause Analysis involves complex causal chains. And the range of Ishikawa Diagram software is not as sophisticated as more generic online collaboration software that relatively easily allows Fault Tree Analysis.