

# What does the reliability of a K out of N system look like?

---

... and what can you do when you see it?



## Workbook

Presented by

**Dr Chris Jackson**

Reliability engineer

Copyright © 2024 Christopher Jackson

All rights reserved. No part of this publication may be reproduced, distributed or transmitted in any form or by any means, including photocopying or other electronic or mechanical methods without the prior written permission of the author (Christopher Jackson), except as permitted under Section 107 or 108 of the 1976 United States Copyright Act and certain other non-commercial uses permitted by copyright law.

For permission requests, write to Christopher Jackson using the details on the following pages.

Limit of Liability/Disclaimer of Warranty: While the author has used his best efforts in preparing this document, he makes no representations or warranties with respect to the accuracy or completeness of the contents of this document and specifically disclaim any implied warranties of merchantability or fitness for a particular purpose. No warranty may be created or extended by sales representatives or written sales materials. The advice and strategies contained herein may not be suitable for your situation. You should consult with a professional where appropriate. The author shall not be liable for any loss of profit or any other commercial damages, including but not limited to special, incidental, consequential, or other damages.

# 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](http://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

<b>System Reliability Modelling</b> .....	<b>5</b>
what is RELIABILITY? (... a revision) .....	5
the RELIABILITY curve.....	7
what does a SYSTEM RELIABILITY MODEL do? .....	9
so .... how does a FAULT TREE work?.....	10
Let's start with an 'AND gate' .....	11
And now the 'OR gate' .....	13
... then there are RELIABILITY BLOCK DIAGRAMS (RBDs) .....	14
<b>'k' out of 'n' system reliability</b> .....	<b>16</b>
Analyzing a 'k OUT OF n' system.....	17
what does '!' mean? ... (FACTORIALs) .....	18
System reliability analysis summary .....	19
'2 out of 3' component system worked exercise .....	20
... so let's look at what 'K OUT OF N' systems do (and don't do).....	<b>21</b>
... the 'PRACTICAL' benefits of REDUNDANCY .....	23
... reliability curves and things like WARRANTY .....	25



## System Reliability Modelling

The term **system reliability** is usually used to separate 'overall product or process' **reliability** from the **component reliability** of its individual elements. We typically first study and understand **component reliability** to understand **system reliability**.

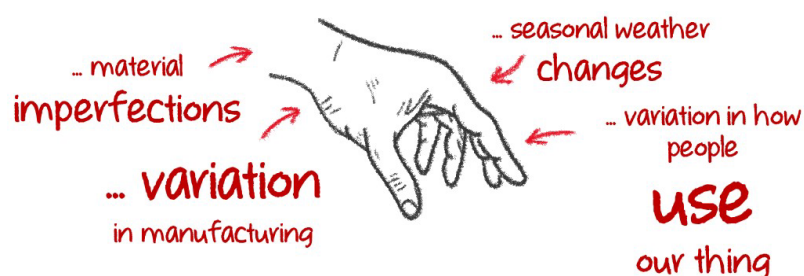
A good understanding of **system reliability** can really help **decision-making**. For example, we can use **system reliability** to determine the optimal **warranty period** for a product we are manufacturing. We don't want too many **failures** within the **warranty period** (this will cost us money and erode profit). We don't want too few **failures** within the **warranty period** (this means we missed an opportunity to advertise a longer **warranty period** to increase market share).

**Reliability** is one way of characterizing the uncertainty of the **failure** process. But just because **failure** is random, doesn't mean it is unpredictable and cannot be used to help **decisions** (such as the **warranty period** decision in the previous paragraph).

what is RELIABILITY? (... a revision)

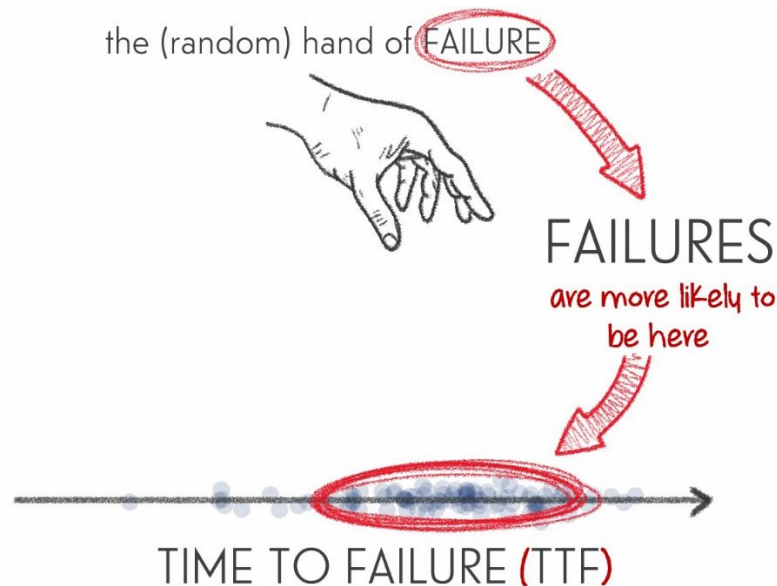
**Random** processes with seemingly identical **inputs** will create different **outputs**. The **random failure process** means that seemingly identical products or systems will **fail** at different times. We might theoretically know that there is underlying microscopic differences from product to product, or that users will use the products in different ways. If these differences are something we cannot discern during production, then each product or system is practically identical to our eyes.

Let's represent some of the factors that introduce uncertainty into the **Time to Failure (TTF)** of a product or system with the **(random) hand of FAILURE** (below). There are hundreds more factors than the four listed below.



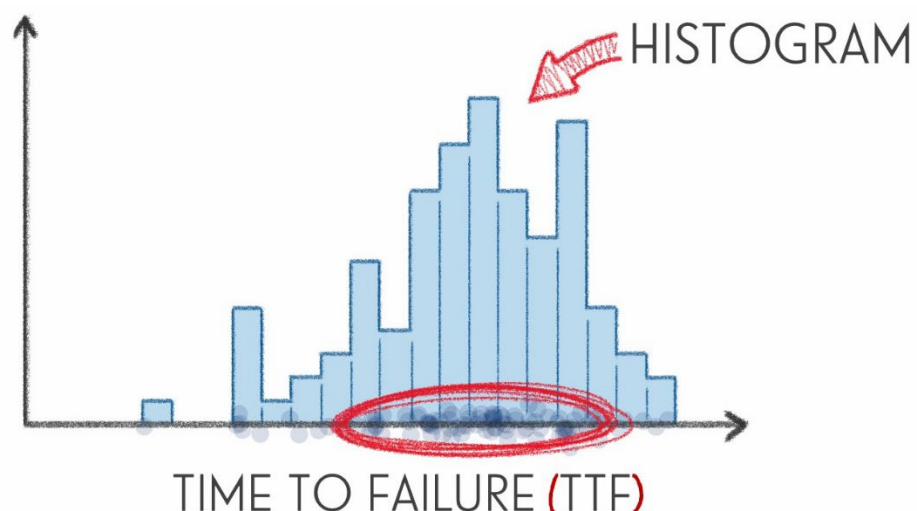
The '**time**' in **TTF** could be measured in terms of distance (mile or kilometers), operating hours, cycles, or any other metric that best captures the age of the system or product. This is often not calendar time.

Even though seemingly identical products might **fail** at different times or ages, there is usually some 'consistency' or 'shared characteristics' in their **TTFs**. Identical models of a particular product might all **fail** around the 'same' time or age, as shown below (each blue circle represents the **TTF** of an individual product.)



Products or systems that tend to **fail** at around the same age or time are **wearing out**. They are slowly accumulating **damage** throughout usage that makes them increasingly likely to **fail**. **Damage** can be many things (like crack length, amount of contaminants in lubrication, diffusion in a semiconductor or corrosion).

While we can see that the blue circles above cluster around a central area creating regions of higher **failure density**, it is not easy to visualize 'how dense' **TTFs** are. This is where a **histogram** can help.



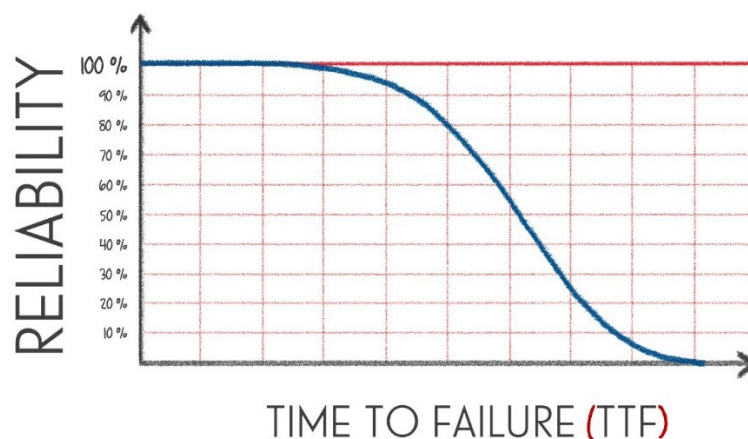
The height of each **bar** in a **histogram** represents the amount or '**density**' of the **TTFs** that fall within the base (or width) of each **bar**. The higher the **bar**, the denser the **TTFs**. **Histograms** really help us visualize the **random** nature of things like **failure**.

**Histograms** help us visualize the **random** nature of **failure**. But a **histogram** doesn't visualize **reliability**. Visualizing **reliability** involves introducing a vertical axis that represents the fraction or percentage of products that have not **failed**. Plotting the **TTFs** on this chart now creates a relatively smooth line.



the **RELIABILITY** curve

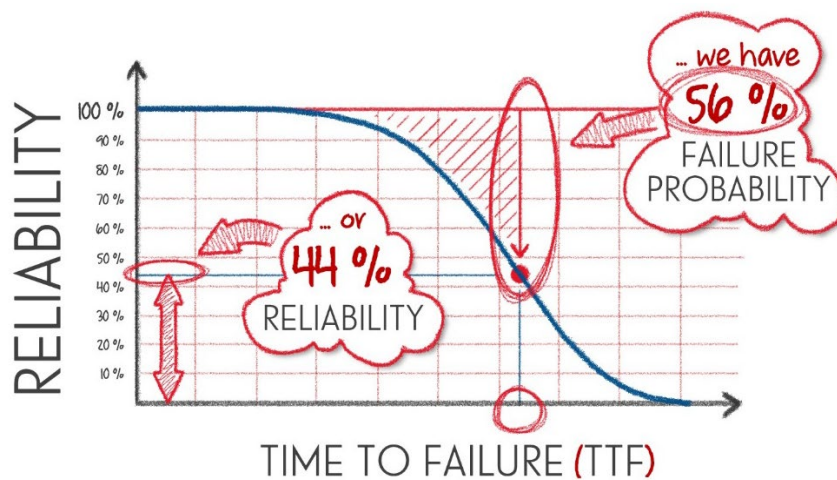
The chart above visualizes the same characteristics of the random failure process, albeit in a different way when compared to the **histogram**. If we were to (hypothetically) observe an infinite number of data points, then a perfectly smooth line will eventually be traced out. This curve is **reliability**.



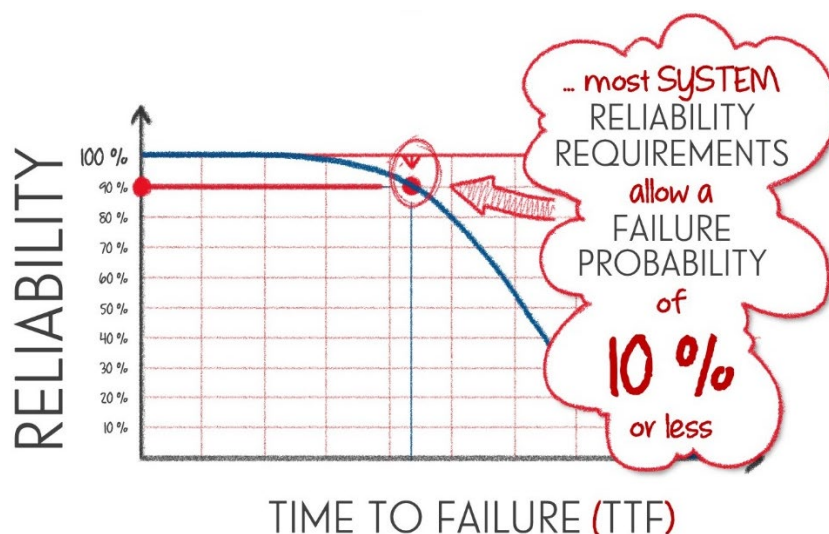
*RELIABILITY is the PROBABILITY that our product HAS NOT FAILED at a particular DURATION in SPECIFIED CONDITIONS*

The **reliability curve** contains a tremendous amount of information regarding the **failure** of our product. Importantly, the curve shows that **reliability** can never be characterized by a single number as it changes over time or usage. The curve above starts at 100 % which implies that a product or system is working at the start of its useful life. The curve that decreases to 0 % also shows that a product or system will always eventually **fail**.

We can use our **reliability curve** to generate a lot of useful information that might help us make better decisions. We can pick any **TTF** and then identify the **reliability** (or **failure probability**) at that point in usage, as illustrated below.

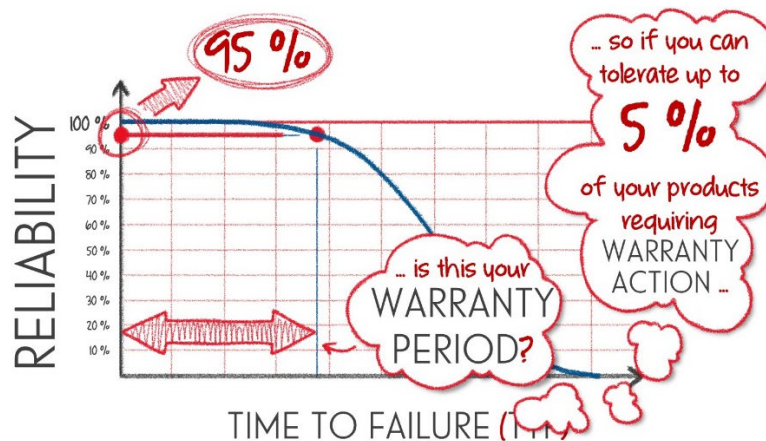


We are usually not interested in the point in time where we expect **reliability** to be anywhere near 50 %. By the time 50 % (or half) of our products have **failed**, they are too old to economically maintain. In other words, they are **obsolete**. We are usually more interested in focusing on the point in usage where a small minority of products or systems have **failed**. Most **reliability requirements** allow for **failure probabilities** of 10 % or less.





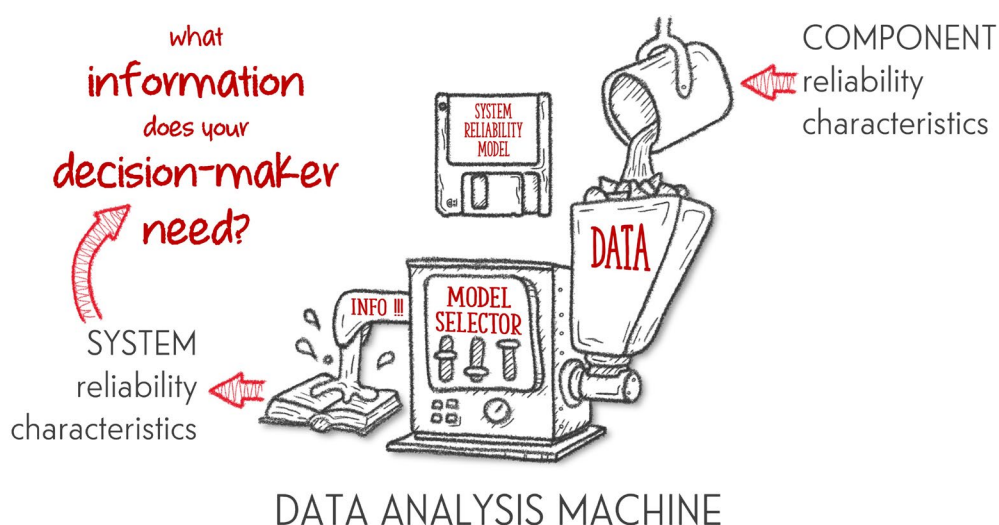
For example, say you were able to work out that for your amazing new product, you can tolerate up to 5 % of your entire sales failing during the **warranty period**. You don't want any more than 5 % to fail otherwise you will start losing too much money through **warranty costs**. You ALSO don't want your **warranty period** to be too short because this won't be attractive to your customers. So we can use our **reliability curve** to help us find the 'best' **warranty period** ...



what does a SYSTEM RELIABILITY MODEL do?

Knowing **system reliability characteristics** helps organizations make better **decisions**. These **decisions** can be the difference between profitability and insolvency. For militaries, understanding **reliability** helps them understand how many spares they will need to win the battle.

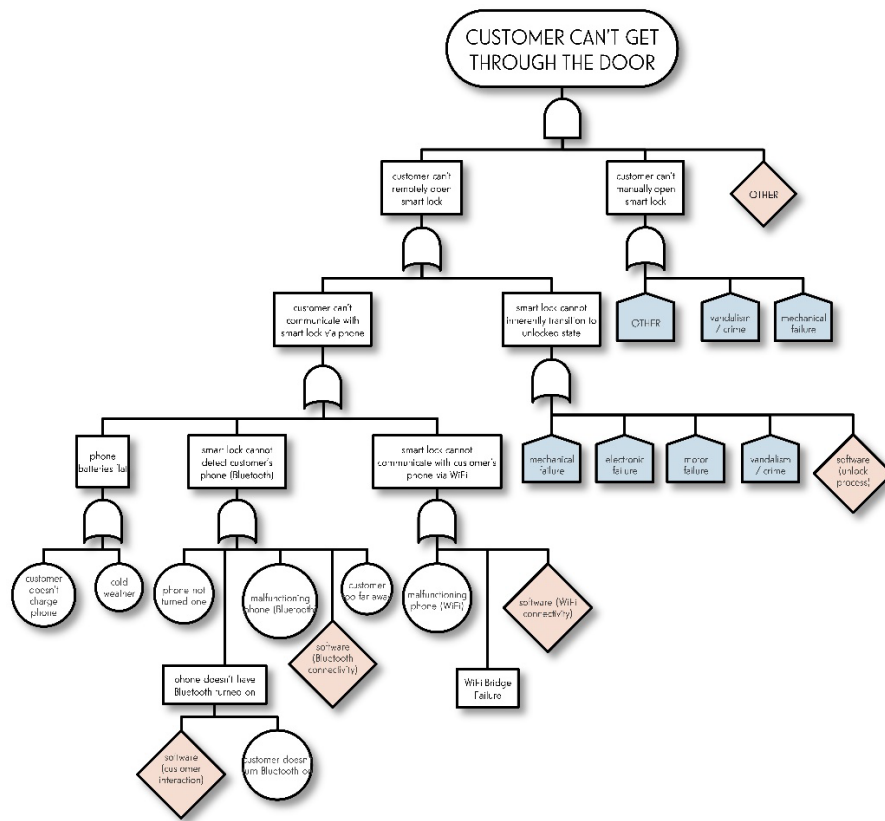
A **system reliability model** allows us to understand **system reliability characteristics** after we understand **component reliability characteristics**. The **system reliability model** encodes the system structures, layouts, configuration and how each component interacts with all others. The **system reliability model** then allows **data analysis** to generate information that SHOULD help a **decision maker** make a **decision**.



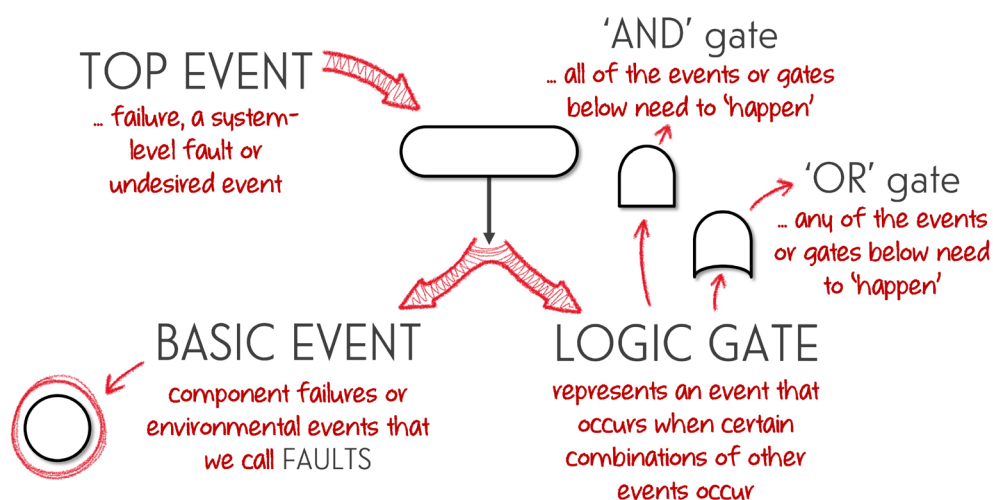
A **fault tree** is one of many ways of representing a **system reliability model**. But to do that, we first need to understand how a **fault tree** works.

## so .... how does a FAULT TREE work?

The example **fault tree** from chapter 1 is illustrated below with the shapes (and the small writing inside each of them) being explained later in this book.



Complicated shapes (like the one above) can be relatively easily generated 'one step at a time.' The most common shapes required to create a **fault tree** are illustrated below.



Most **fault trees** are built by starting with the **top event** which is depicted by a thin horizontal shape with round ends. This represents the **undesirable event** around which is the focus of the **fault tree**. A line is drawn down from the **top event** to either a single **basic event** or a **logic gate**.

A **basic event** is often referred to as a **system fault**, which is where the **fault tree** gets its name. **Basic events** are depicted by circles or ovals. A **basic event** in a **system reliability model fault tree** typically represents **component failure**. In other types of **fault trees**, **basic events** can be design flaws, environmental events, human errors, or any other scenario that might contribute to **failure**.

**Logic gates** represent an **event** which is the **combination of other events**. This combination can include **basic events** and other **logic gates**.

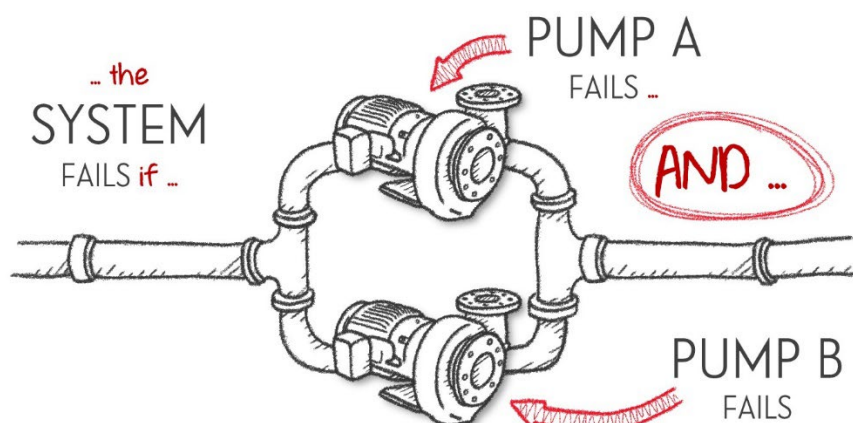
The **logic gate** depicted by the shape with a round top and a flat bottom is an '**AND gate**.' An '**AND gate**' represents a combination where ALL the **events** or **gates** 'beneath' it need to occur for the **fault** to propagate to the **top event**. The **events** and **gates** 'beneath' the '**AND gate**' are called **input events**.

The **logic gate** depicted by the shape with a round top and round (inverted) bottom is an '**OR gate**.' An '**OR gate**' represents a combination where AT LEAST ONE **input event** needs to occur for the **fault** to propagate to the **top event**.

The '**AND gate**' and '**OR gate**' are the two most common **logic gates** used in **FTA**. Other **logic gates** used to create **fault trees** are examined later.

Let's start with an 'AND gate'

Consider the simple, two pump system below.



The system **fails** if pump A fails **AND** pump B **fails**. In other words, we need at least pump to **function** for the system to **function**. This is called a **parallel system** as the **components** are often configured on 'parallel operating lines' like pumps to allow the system to **function** if at least one component is **functioning**.

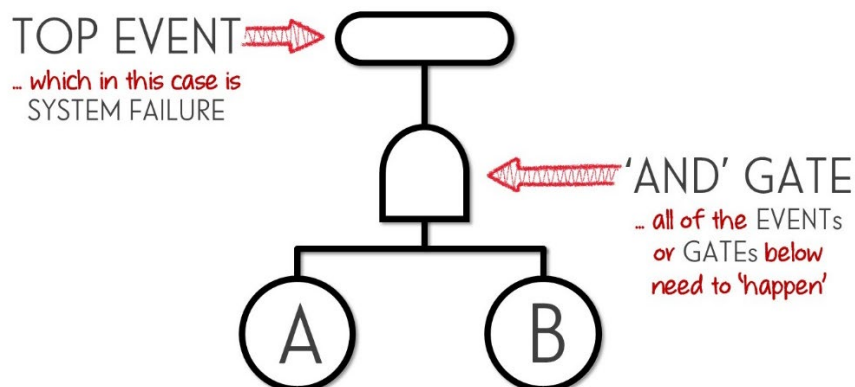
*The most important word for this system is 'AND.'*

To create a **fault tree** that models **two-pump parallel system reliability**, we start with defining the **basic events**.

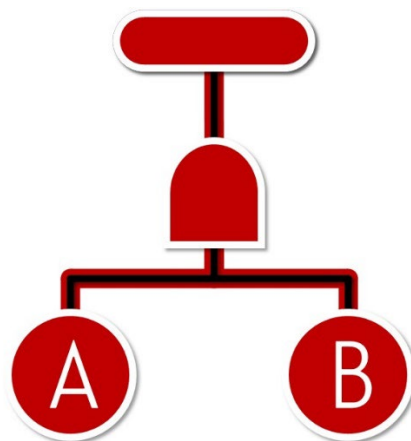
*Basic event 'A' is the event where pump A fails.*

*Basic event 'B' is the event where pump B fails.*

Because the system **fails** if pump A **fails AND** pump B **fails**, we need use an '**AND gate**' to create the **two-pump parallel system reliability fault tree**.



To simplify the **fault tree** illustration, we use **component identifiers** when depicting **basic events**. The letters 'A' and 'B' in the **fault tree** above mean the **basic events** represent the **failures** of pump A and pump B respectively. So for the **two component parallel system, failure** looks like this (where red represents **failure**), meaning both pumps have **failed** ...

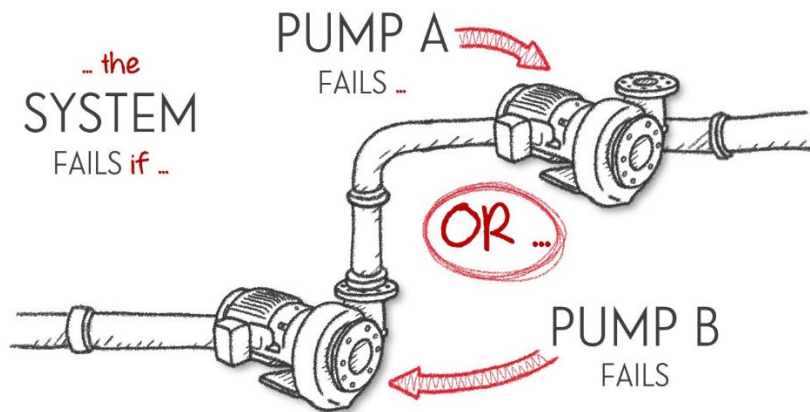


**Parallel systems** involve **redundant components** that can ensure the system remains **functional** when one of the other (usually identical) **components fail**.



## And now the 'OR gate'

Let's look at a different two-pump system. This system needs one pump to pump fluid 'up' to a level that is higher off the ground. The second pump then pumps that fluid from the 'higher' place somewhere else.



The system **fails** if pump A **fails** **OR** pump B **fails**. This is called a **series system** (mainly because our pumps are arranged one after the other in **series**).

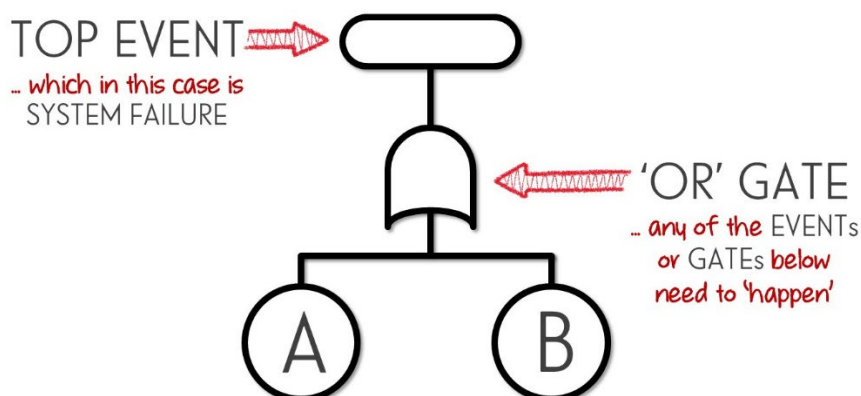
*The most important word for this system is 'OR.'*

The two **basic events** are the same as those for the two-pump parallel system as the components have not changed.

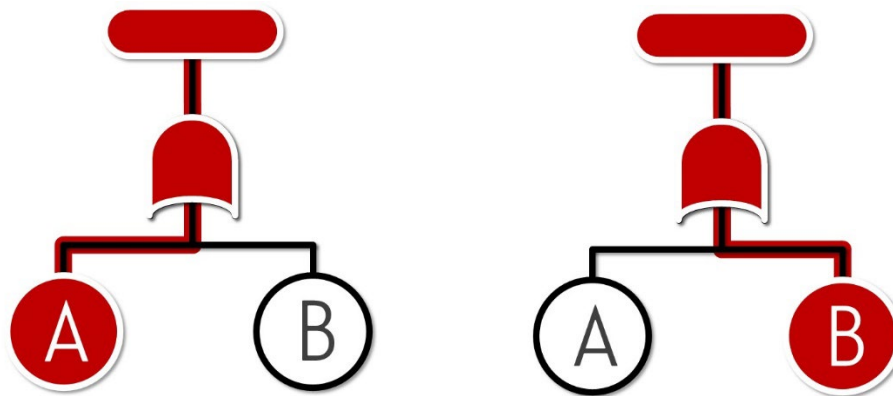
*Basic event 'A' is the event where pump A fails.*

*Basic event 'B' is the event where pump B fails.*

This allows us to use an '**OR gate**' to create the following **two-pump series system reliability fault tree**.

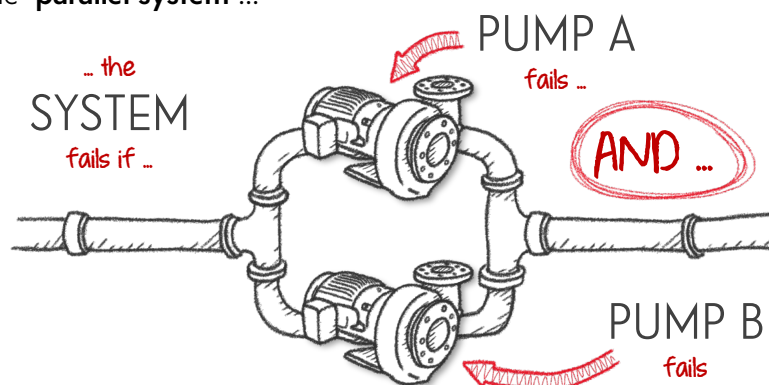


So for the **two-pump series system fault tree**, the two **failure** scenarios are represented below (where red represents **failure**) ...

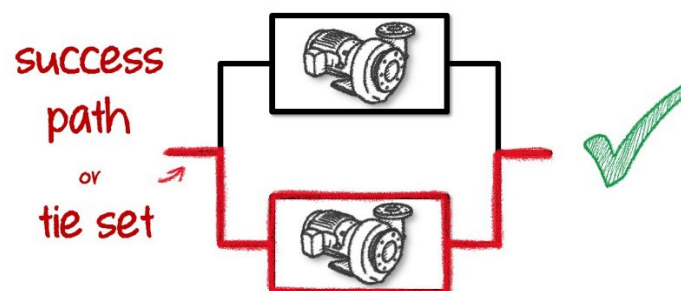


... then there are RELIABILITY BLOCK DIAGRAMS (RBDs)

Going back to the **parallel system** ...

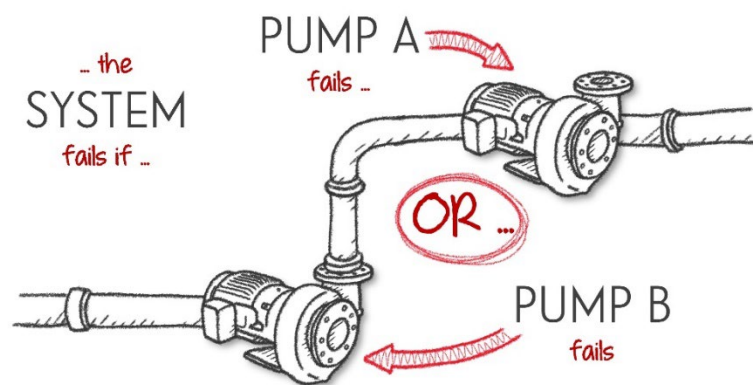


Another way we can model the **reliability** of this simple parallel system is with a **Reliability Block Diagram (RBD)**.



**RBDs** represent all components with blocks that are joined together with lines. It allows us to understand which combinations of components need to be functional for the system to work. If we can draw a line from one side of our **RBD** to the other ... **AND** that line passes through blocks (components) that have not failed, then the system is functional. The line is called a **success path** or **tie set**.

Going back to the **series system** ...



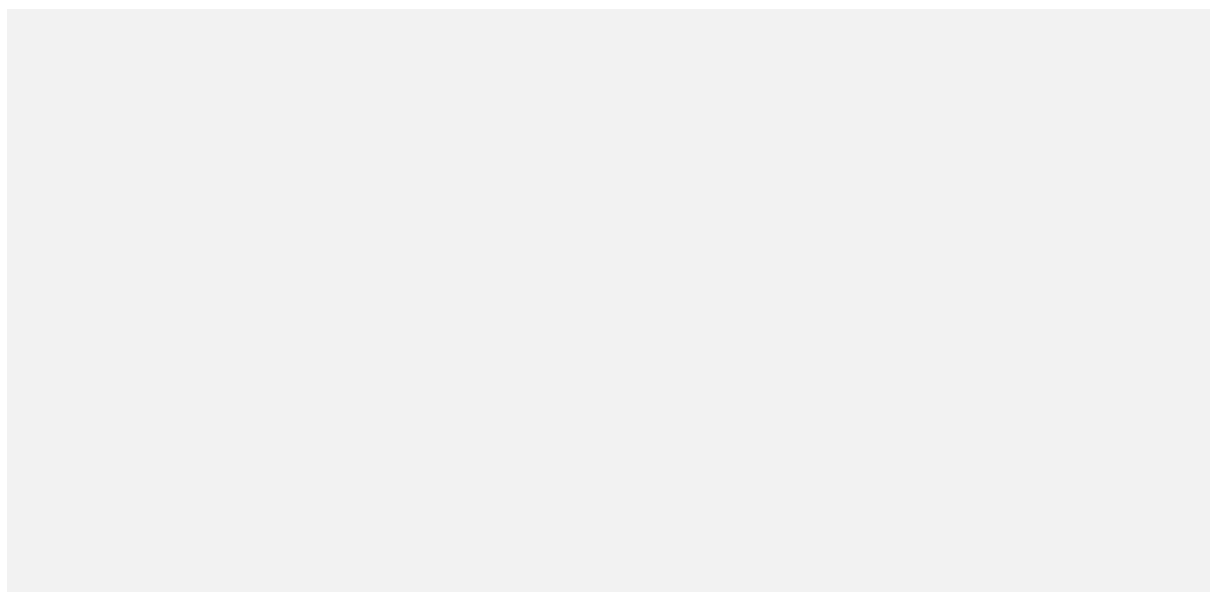
The corresponding **RBD** for this **series system** is illustrated below.



**RBDs** and **fault trees** can represent the same system equally well, depending on what you are trying to represent. **RBDs** can mimic the physical layout of the components in your component or sub-system. But **RBDs** can usually only incorporate **component failure events**. **Fault trees** do not mimic the physical layout of a system but can align with how our brains like to analyze system reliability.

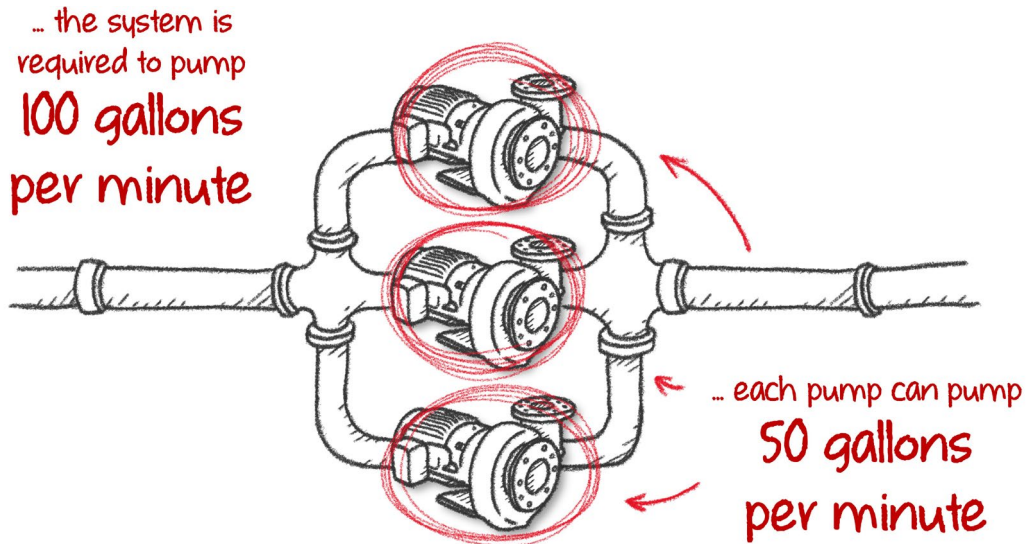
Use the methodology that works best for you.

But most systems have more than two components. **System reliability model fault trees** are usually more complicated than the simple **series** and **parallel** examples above. However, the approach to creating a more complicated **fault tree** is still relatively straightforward if approached in a methodical way.



## 'k' out of 'n' system reliability

Many systems require logic beyond simple '**AND**' and '**OR gates**' to fully represent how they work. Look at the three-pump system below.



With each pump being able to pump 50 gallons per minute, and the system requiring 100 gallons per minute to be pumped, we need two pumps to be functional. This means we can tolerate one pump **failing** (in other words, one pump is **redundant**). So the system is not a **series system**.

However, we need more than one pump to work for the system to meet its 100 gallons per minute requirement. So the system is not a **parallel system**.

This is what we call a '**k out of n**' system. '**k**' refers to the minimum number of components that need to be working for the system to work. '**n**' refers to the total number of components that have the 'ability' to function.

Our three-pump system above is a '**2 out of 3**' system. '**k out of n**' systems are represented using a special logic gate symbol that resembles an '**OR gate**' symbol, but contains the number of functional systems for it the gate to NOT be true.

*In other words, we need at least 'k' things to be functional for our subsystem to be functional.*

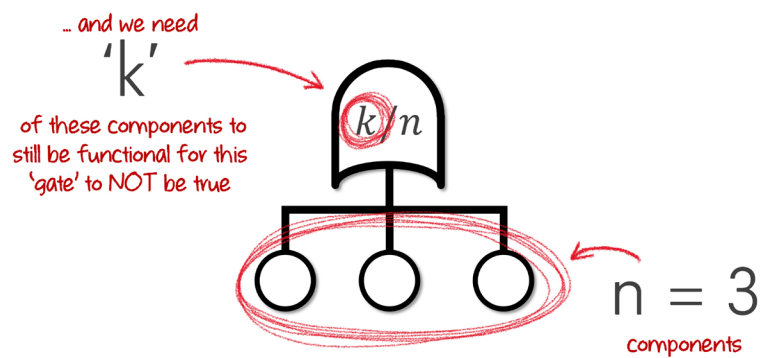


**'k' OUT OF 'n'**

... where we need at least '**k**' events to occur or happen

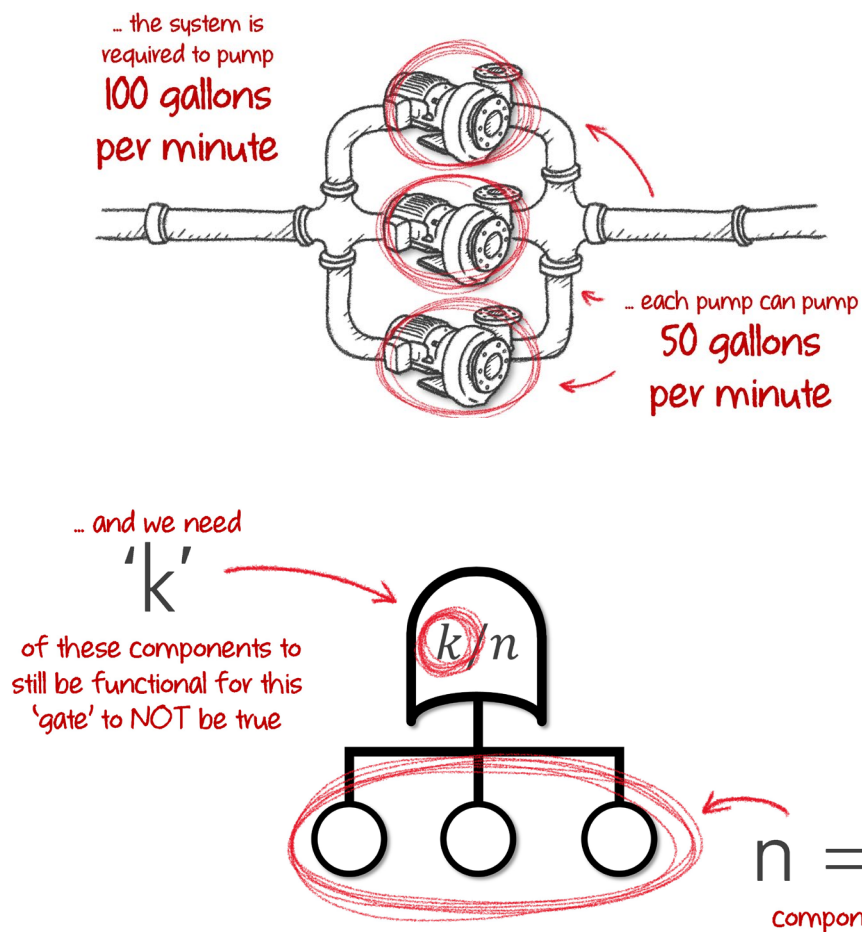


So it is important to ensure that for any '**k out of n**' gate, there must be '**n**' input events.



### Analyzing a '**k OUT OF n**' system

Recall our '**2 out of 3**' pump system, which is a form of a '**k out of n**' system.



The equation to calculate 'k out of n' system reliability is a little more complicated. If the reliability of each component is the same (or in other words, all components are the same), then the 'k out of n' system reliability becomes ...

$$R_{System}(t) = \sum_{i=k}^n \frac{n! [R(t)]^i [1 - R(t)]^{n-i}}{i! (n-i)!}$$

... this is how we sum for all these equations from 'i' equals 'k' all the way to 'n'

This equation would usually need software to solve it. And if the components have different **reliabilities**, then the expression above needs to be derived 'by hand' using potentially complicated **Boolean algebra**. Fortunately, most 'k out of n' systems involve the same components, meaning they have the same **reliabilities**, which then allows us to use the equation above.

what does '!' mean? ... (FACTORIALs)

You might have noticed that there was a strange symbol in the 'k out of n' system reliability equation on the previous page. Some of you might know what this symbol means (in a mathematical context). Some of you might not. So which symbol are we talking about?

$$R_{System}(t) = \sum_{i=k}^n \frac{n! [R(t)]^i [1 - R(t)]^{n-i}}{i! (n-i)!}$$

We are talking about the '!' which some people call the **exclamation point** or **exclamation mark** when used in a sentence. But when we see the '!' symbol in a mathematical formula, it is called the **factorial**.

So what is a **factorial**? It is simply the product of all the integers starting from 1 up to a certain number. For example 'n factorial' or 'n!' is defined as ...

$$n! = n \times (n - 1) \times (n - 2) \times \dots \times 3 \times 2 \times 1$$

For (another) example, '3 factorial' or '3!' is ...

$$3! = 3 \times 2 \times 1 = 6$$

And some more examples ...

$$4! = 4 \times 3 \times 2 \times 1 = 24$$

$$2! = 2 \times 1 = 2$$

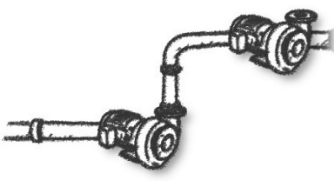
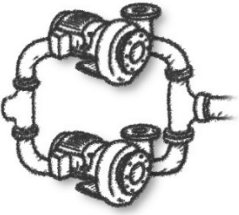
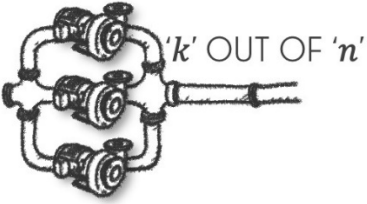
The **factorial** is very useful when we need to deal with **combinations** of things. In a '**k out of n**' system, we need to deal with a combination of '**k**' functional components out of a total of '**n**' components. That is why we see '**!**' in the reliability function of a '**k out of n**' system.

**Factorials** can become very difficult to calculate once the integers involved start getting bigger. So as a rule, you should use a software tool like Microsoft Excel to help you out. For example, you can use the following formula in Microsoft Excel to calculate '**4 factorial**' or '**4!**' ...

 = FACT(4)

## System reliability analysis summary

So we have now worked out how to calculate the **reliability** and **failure probability** of **series**, **parallel** and '**k out of n**' systems. The corresponding formulae are summarized below.

$R_{System}(t)$	$\prod_{i=1}^n R_i(t)$	$1 - \prod_{i=1}^n 1 - R_i(t)$	$\sum_{i=k}^n \frac{n! [R(t)]^i \times [1 - R(t)]^{n-i}}{i! (n-i)!}$
$F_{System}(t)$	$1 - \prod_{i=1}^n 1 - F_i(t)$	$\prod_{i=1}^n F_i(t)$	$\sum_{i=k}^n \frac{n! [R(t)]^i \times [1 - R(t)]^{n-i}}{i! (n-i)!}$

It starts with being able to recognize what characteristics of a **system** make it '**series**,' '**parallel**,' or '**k out of n**.' This allows you to create the right **system reliability model** using a tool like a **fault tree**. The next two chapters involve us practicing how to use these formulae for the systems we have examined so far. So by the end of chapter 7, you should be an expert in analyzing system reliability!

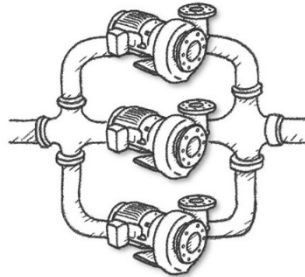
## '2 out of 3' component system worked exercise

### EXERCISE

Calculate the reliability of the following '2 out of 3 pump system' at 2 years.

The reliability of each pump at 2 years is 0.7 or 70%.

OUT OF



$$R_{System}(\text{ }) = \sum_{i=0}^{\text{ }} \frac{\text{ }! [\text{ }]^i [1 - \text{ }]^{\text{ } - i}}{i! (\text{ } - i)!}$$

$$= \frac{\text{ }! [\text{ }]^{\text{ }} [1 - \text{ }]^{\text{ } - \text{ }}}{\text{ }! (\text{ } - \text{ })!} + \frac{\text{ }! [\text{ }]^{\text{ }} [1 - \text{ }]^{\text{ } - \text{ }}}{\text{ }! (\text{ } - \text{ })!}$$

Now ... evaluate all the subtractions

$$= \frac{\text{ }! [\text{ }]^{\text{ }} [\text{ }]}{\text{ }! (\text{ })!} + \frac{\text{ }! [\text{ }]^{\text{ }} [\text{ }]}{\text{ }! (\text{ })!}$$

Now ... evaluate all the factorials ...

$$= \frac{\text{ } [\text{ }]^{\text{ }} [\text{ }]}{\text{ } \times \text{ }} + \frac{\text{ } [\text{ }]^{\text{ }} [\text{ }]}{\text{ } \times \text{ }}$$



# EXERCISE

Now ... simplify the fractions ...

$$= \boxed{\phantom{00}} \left[ \boxed{\phantom{00}} \right] \boxed{\phantom{00}} \left[ \boxed{\phantom{00}} \right] \boxed{\phantom{00}} + \boxed{\phantom{00}} \boxed{\phantom{00}}$$

Now ... substitute in the value for pump reliability at 2 years (if you haven't already done so) -  $R(2)$  ...

$$= \boxed{\phantom{00}} \left[ \boxed{\phantom{00}} \right] \boxed{\phantom{00}} \left[ \boxed{\phantom{00}} \right] \boxed{\phantom{00}} + \boxed{\phantom{00}} \boxed{\phantom{00}}$$

Now ... evaluate the terms each side of the '+' sign ...

$$= \boxed{\phantom{00}}$$

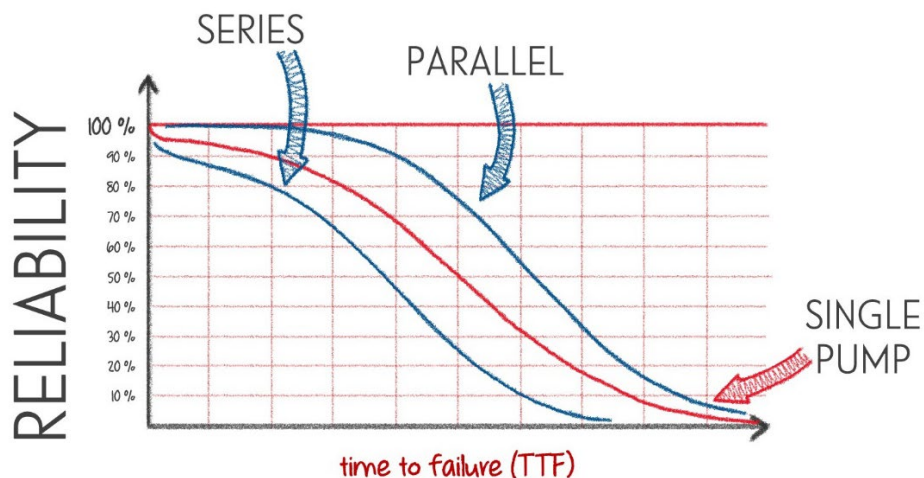
$$+ \boxed{\phantom{00}}$$

Now ... solve the equation!

$$= \boxed{\phantom{00}}$$

... so let's look at what 'K OUT OF N' systems do (and don't do)

In chapter 6, we calculated **two-parallel component** and **two-series component system reliabilities** and compared them to the **reliability curve** of a single component. The **series system**, **parallel system** and component **reliability curves** for the pumps we have been examining are illustrated below.

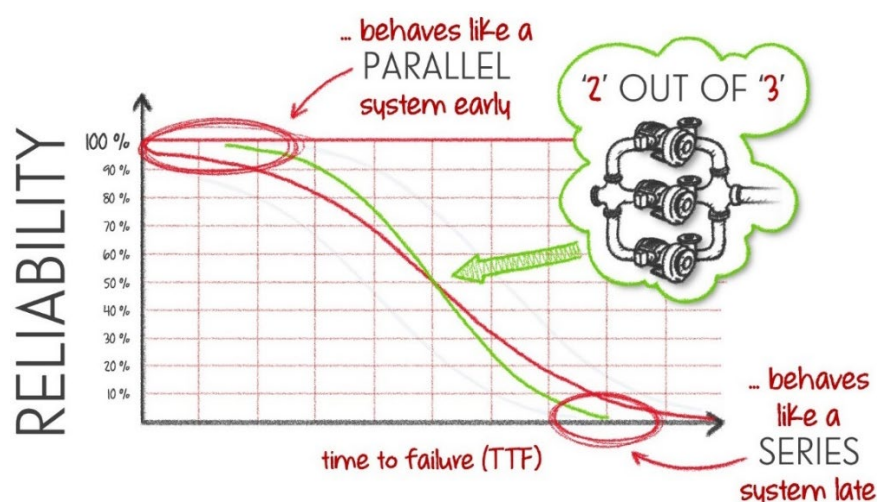


**Parallel systems** are MORE **reliable** as they include **redundant components** that provide more ways for the system to **WORK**. **Series systems** are LESS **reliable** as they include more '**single point failure**' components that provide MORE ways for the system to **FAIL**.

'**k out of n**' systems incorporate characteristics of both **series** and **parallel systems**. Let's think about a '**2 out of 3**' system. When the '**2 out of 3**' system is 'young,' it is more likely to have all three components still working. This means that it is more likely to have one **redundant component**. This **redundancy** makes it similar to a **parallel system**.

But when the '**2 out of 3**' system is 'old,' it is less likely to have all three components still working. This means it is more likely to have no **redundant components**. This lack of **redundancy** makes it similar to a **series system**.

And this is what we observe in the '**2 out of 3**' system **reliability curve** below. You can see that when the '**2 out of 3**' system is 'young,' its **reliability curve** is like the **two-parallel component system reliability curve**. And when the '**2 out of 3**' system is 'old,' its **reliability curve** is like the **two-series component system reliability curve**.



This is a really important observation. When it comes to **reliability**, we typically focus on the life of a system when its **reliability** is still greater than (for example) 90 %. Having 10 % or more of your systems **fail** means that you have taken too long to retire them, have offered a warranty period that is too long, or haven't conducted **Preventive Maintenance (PM)** as often as you should be.

This means that a '**2 out of 3**' system can offer similar **reliability** performance as a **two-parallel component system**. And, we now have a '**load sharing**' system. If (for example) our pumping system needs to pump 100 gallons per minute, then a **two-parallel pump system** and '**2 out of 3**' **pump system** will offer similar **reliability** performance when 'young.' But, the '**2 out of 3**' system only needs to have pumps with 50 gallon per minute capacities as two pumps are allowed to combine to meet the system requirements. The **two-parallel pump system** must have bigger, more expensive 100 gallon per minute pumps to have **reliability characteristics** similar to our '**2 out of 3**' **pump system**. This gives designers an opportunity to save money without sacrificing reliability.

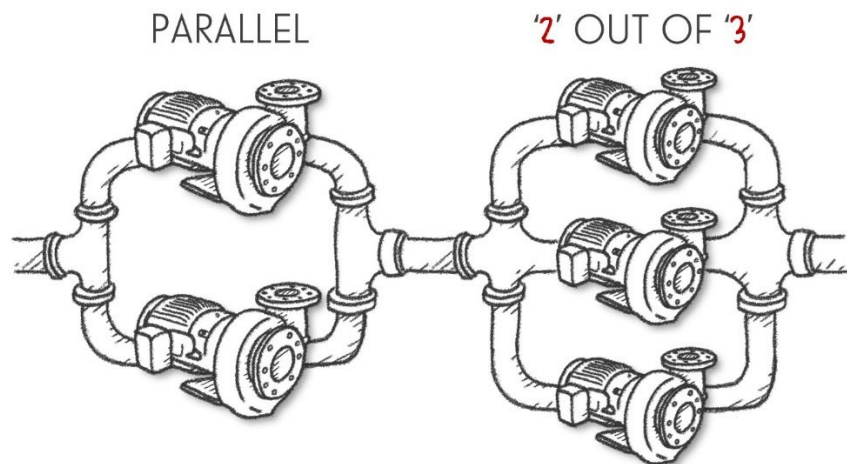
## ... the 'PRACTICAL' benefits of REDUNDANCY

Both parallel and 'k out of n' systems incorporate redundancy. This increases system reliability by providing more ways the system can work. But in addition to improved reliability, there are more 'operational' benefits for having redundancy in your system. Can you think what that might be?

### EXERCISE

A pumping system comprises two 'modules:' one module is a two-parallel pump subsystem, and the other module is a '2 out of 3' pump subsystem.

Both modules are arranged in series.

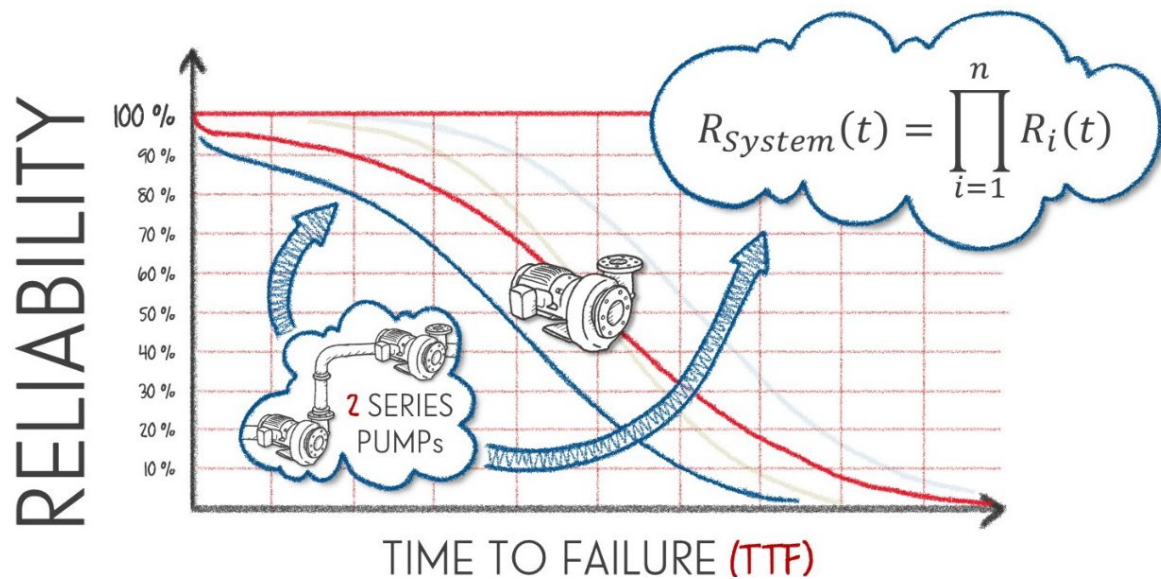


What is the **ADDITIONAL** benefit of any system that incorporates **REDUNDANCY**? Think about what you can **DO** to your system if one pump fails.

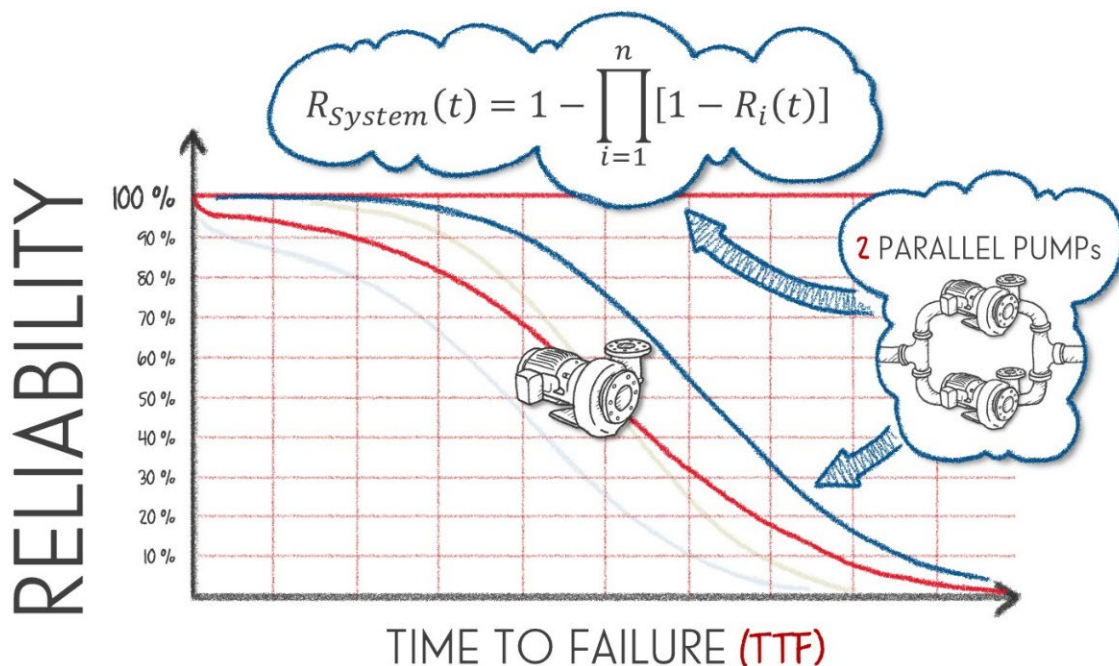
So **redundancy** can be very useful for both **reliability** and operations. The examples we have looked at in Chapters 6 and 7 have all focused on 'point estimates' of **reliability** at specific times. Of course, we can use these equations to generate entire **reliability curves**, especially if we know what the **reliability curve** of each component is. There is so much information we can 'pull' from a **reliability curve** ... and that will be the focus of the next chapter.

The previous two chapters involved us practicing how to evaluate **system reliability** if we know **component reliabilities** for **series**, **parallel** and '**k out of n**' systems. This approach can be applied to all reliability values of a component throughout its useful life. For example, if the reliability of a single pump is represented by the red line below, then we can use the equation below to create a **reliability curve** for a **two-series pump system**. As you would expect, because **series systems**

effectively introduce MORE ways your system can fail, the **two-series pump system reliability curve** is lower than the pump **reliability curve**.



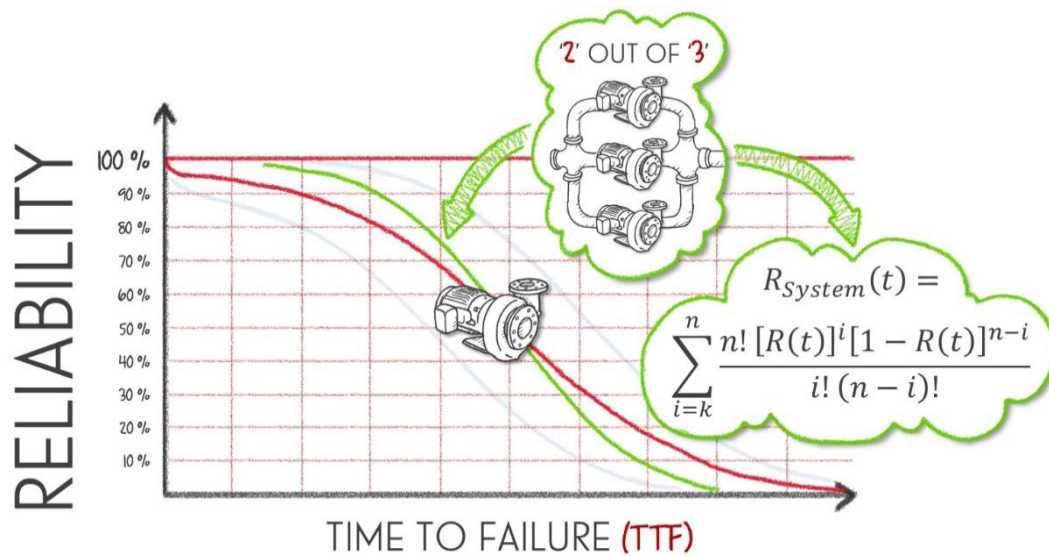
As opposed to a **series system**, a **parallel system** involves **redundancy**. This means that we effectively introduce MORE ways our system can work which improves **reliability**. The **two-parallel pump system reliability curve** (below) is higher than the pump **reliability curve**.



And finally, when we analyzed a '**2 out of 3**' system, we get reliability characteristics of both **series** and **parallel systems**. When a '**2 out of 3**' system is 'young,' it is more likely to have all three components working. This means that it is likely to have one **redundant** component, which is similar to a **two-parallel component system**. But when a '**2 out of 3**' system is 'old,' it is less likely to have all three components working. This means that it is likely to have no **redundant** components, which is similar to a **two-series component system**. This can be seen below in the green '**2 out of 3**' pump series system reliability curve. The green line is 'close' to the **two-parallel**



**pump system** when it is young (when reliability is high). The green line is 'close' to the **two-series pump system** when it is old (when **reliability** is low).



These **reliability curves** might be 'nice' to look at in terms of the way they represent information ... but they are useless if they don't influence a decision. So how can we use them?

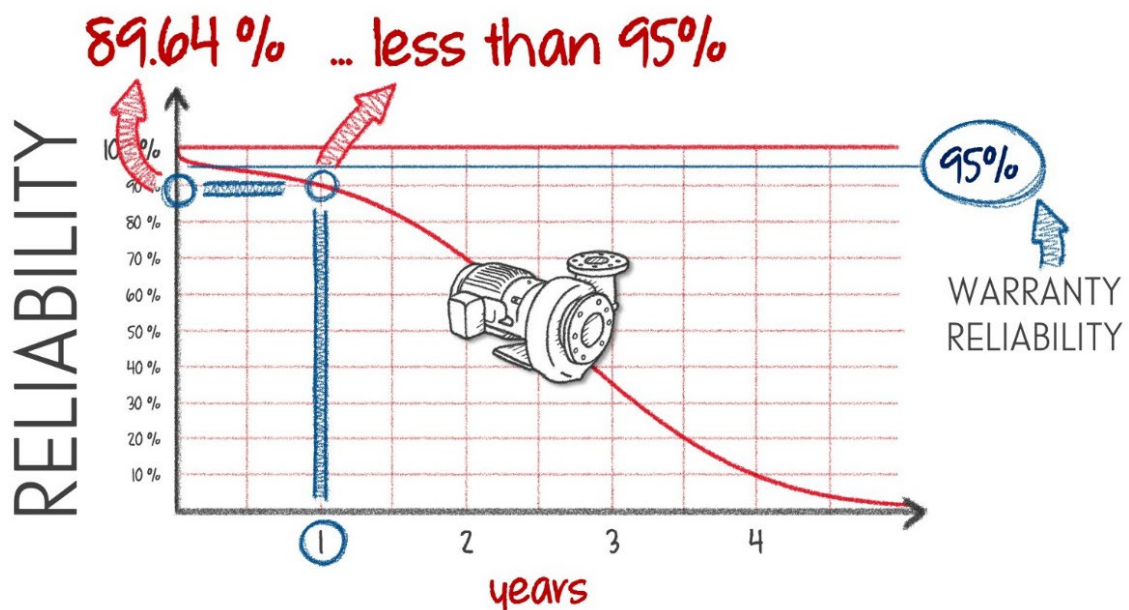
... reliability curves and things like WARRANTY

Let's keep focusing on a pump system which has the following characteristics or requirements ...

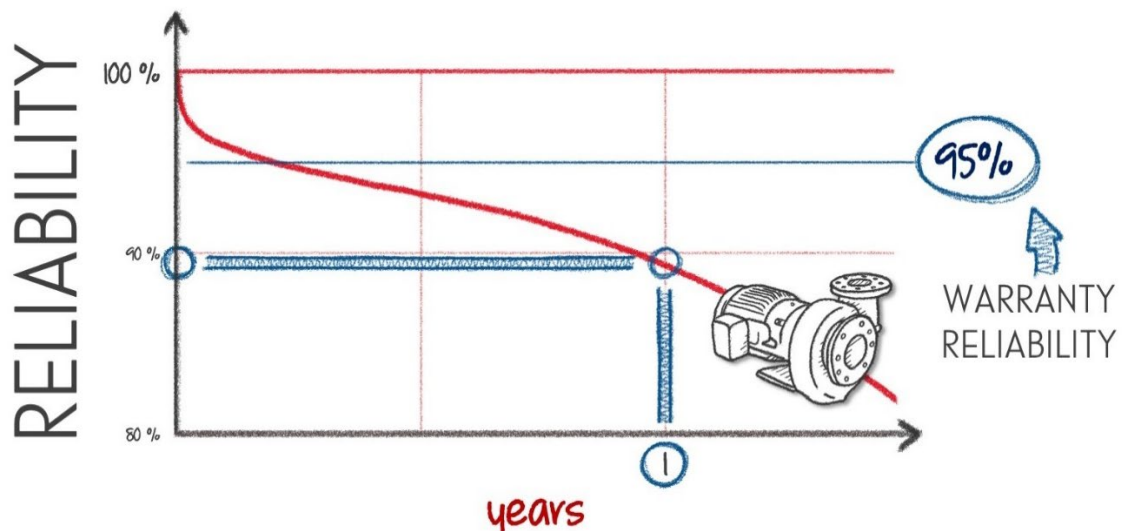
... must be able to pump a total of 100 GALLONS PER MINUTE ...

... must have a ONE YEAR WARRANTY RELIABILITY of 95 % ...

If we focus on the single 100-gallon per minute pump **reliability curve** (in red below), it tells us that the **reliability** of a system that contains that single pump is less than 95 % (89.64 %).



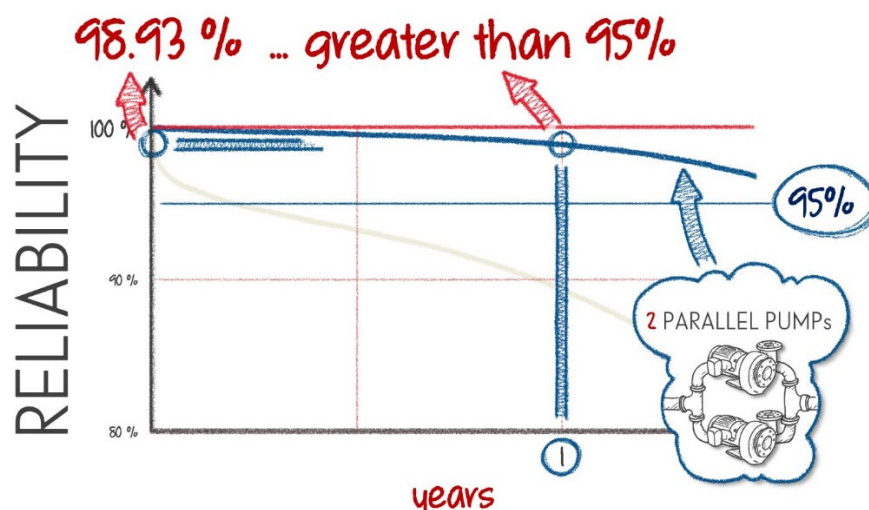
As we can see, the part of the **reliability curve** that we (and most reliability engineers) are interested in is the 'early' part of the life of the pump where **reliability** is still relatively high (around 95 %). The illustration below shows a 'zoomed in' representation of the **reliability curves** above that focuses on the **one-year warranty period**.



So we have a problem if we only have a single 100-gallon per minute pump because its **one-year warranty reliability** (89.64 %) is less than the requirement of 95 %.

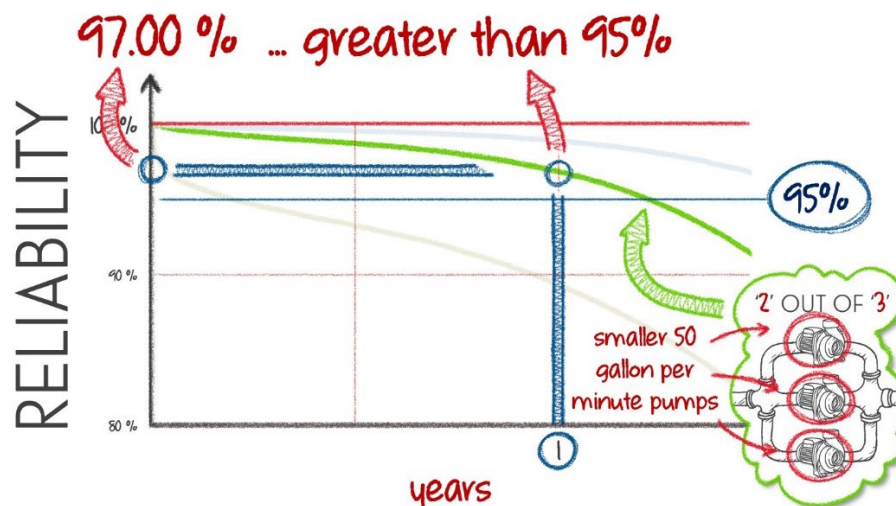
*So what can we do?*

We know that adding a **redundant** pump to make a **two-parallel pump system** increases **reliability**. The **reliability curve** of a **two-parallel pump system** is illustrated below, showing that the **reliability** at one year is 98.93 % ... which exceeds the **warranty period requirement** of 95 %.



But good reliability engineers should look for lots of different options to explore options for saving money and being more efficient. Which is why we should look at **'2 out of 3' system reliability**. Why? Because a **'2 out of 3' system** is able to have two pumps *together* meet the requirement of the 100-gallons per minute. This allows smaller, perhaps less expensive 50-gallons per minute pumps be used (two 50-gallons per minute pumps can pump a combined 100-gallons per minute).

So if there is a 50-gallons per minute pump that has a similar **reliability curve** as the 100-gallons per minute pump we looked at above, then we can create a green '**2 out of 3**' pump system **reliability curve** below that suggests that we have **one-year warranty reliability** of 97.00 %. This is less than the **reliability** of a two 100-gallons per minute parallel pump system of 98.93 % ... but it still meets the requirement of 95 %.



So why do we want *options*? Because flexibility in design options generates money. Opportunities can be identified and explored. You might be able to create a system that is better than that of your competitors.

Maybe using 50-gallon per minute pumps allows them to be installed into 'small spaces' within a plant facility that 100-gallon per minute pumps simply cannot. Or maybe using 100-gallon per minute pumps in a **two-parallel pump system** allows the manufacturer to extend the warranty period.

*Flexibility in design options generates money.*