

# 10 Reasons to do Reliability

---

If reliability engineering isn't making you money, then you are doing it wrong



## Workbook

Presented by

**Dr Chris Jackson**

Reliability engineer



Copyright © 2023 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.

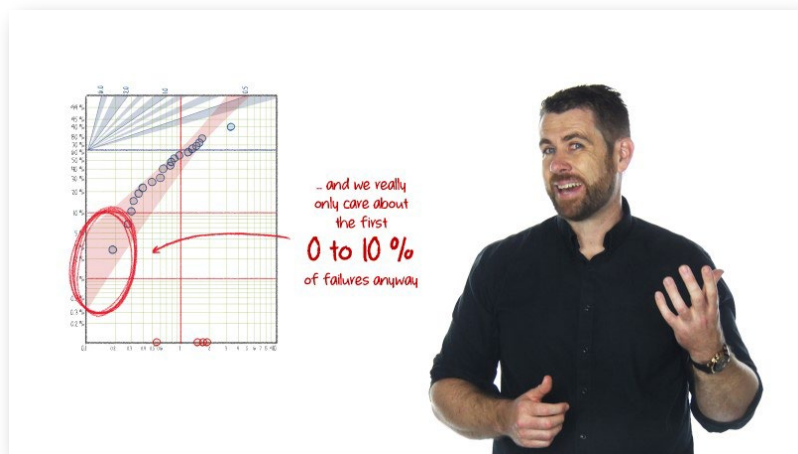
Limit of Liability/Disclaimer of Warranty: While the author has used his best efforts in preparing this document, he makes no representations or warranties with respect to the accuracy or completeness of the contents of this document and specifically disclaim any implied warranties of merchantability 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>10 REASONS TO DO RELIABILITY .....</b>	<b>5</b>
Reason #1 – reliability engineering ≠ reliability purgatory .....	6
Reason #2 – eliminate ALL problems .....	6
Reason #3 – NO complex, expensive fixes .....	10
Reason #4 – Saving (LOTS OF) time and money .....	14
Reason #5 – quickly solving the VITAL FEW .....	16
Reason #6 – Making your thing better (than your competitors).....	17
Reason #7 – Generating VALUE .....	18
Reason #8 – No OVERWHELM.....	21
Reason #9 – happier people .....	22
Reason #10 – your REPUTATION.....	23
It starts with the RELIABILITY MINDSET .....	24



## 10 Reasons to do Reliability

There are lots of different ideas about what **reliability engineering** is. Many feel that **reliability engineering** is ONLY about MEASURING reliability and perhaps finding out why failures happened ... after they happen. Others feel that **reliability engineering** is a set of 'checks and tests' that prevent really bad catastrophes from happening – like the one below ...



This is not **reliability engineering**.

or analyzing  
... measuring <sup>^</sup> reliability CAN be important ...  
... but not nearly as important as DESIGNING IN RELIABILITY ...

For **reliability** to happen, people need to want to do it. So here are 10 reasons why you should be excited about **reliability engineering**.

1. RELIABILITY ENGINEERING ≠ RELIABILITY PURGATORY
2. eliminate ALL PROBLEMS
3. no COMPLEX, EXPENSIVE FIXES
4. saving (LOTS OF) TIME AND MONEY
5. quickly solving the VITAL FEW
6. making your thing BETTER than your competitors
7. generating VALUE
8. no OVERWHELM
9. HAPPIER people
10. your REPUTATION

So let's look at each one of these in more detail to hopefully make you (and everyone you talk to) WANT to do **reliability engineering**.

## Reason #1 – reliability engineering $\neq$ reliability purgatory

Analyzing (or admiring) a problem doesn't solve it. It is just the first step. But when we make **analyzing reliability** so challenging and hard, we can sometimes feel that all effort created an outcome. **Reliability engineering** means **improving** design, manufacturing, or maintenance to generate value through **improved reliability (and availability and maintainability) performance**. Following textbooks are only important some of the time, and for some situations.

One of the reasons people perceive ALL reliability activities to include (painfully) complex analysis activities is because of our **first enemy of reliability**, the ponderous professor.



They often can't tell you why all the stuff they say you need to do is important. They can't help you make a decision quickly. But they will say that **reliability** is very important to prevent catastrophes and 'bad things' in general.

Good **reliability engineering** involves focusing on the decision you are trying to inform and then working out how to get that information. And if you are focusing on improving reliability and not measuring it ... this is often easier than you might think.

## Reason #2 – eliminate ALL problems

(Good) **reliability engineering** eliminates problems. A common misconception is that **reliability engineering** only addresses the following, very costly problem:

**PROBLEM #1** (*catastrophe*) ... YOUR PRODUCT, SYSTEM OR SERVICE STOPS WORKING FOR YOUR USER OR CUSTOMER ... WHICH IS ALSO KNOWN AS **FAILURE**



The next problem that reliability engineering always addresses (when done well) is ...

**PROBLEM #2** (*devastating crisis*) ... PROTOTYPES CAN'T PERFORM OR SOME OTHER ISSUE IS IDENTIFIED DURING MANUFACTURE OR JUST BEFORE LAUNCH



If we only find the problem just before we are supposed to manufacture or launch our product, we may face an impossible dilemma – spend a large amount of money to redo the stuff we thought we had done well ... or knowingly launch an unreliable product.

Many organizations simply ‘hope’ that any problems that are uncovered during manufacture are ‘anomalies’ or ‘outliers’ that won’t ever cause problems in the hands of customers and users. This is wishful thinking and often driven by the fact that budgets and schedules are already exhausted ... so we try and convince ourselves that everything will be fine. But this usually means that not spending lots of money to fix that problem during manufacture means we lose even more money when it **fails** in the hands of the user or customer.

So it is usually too late by the time we stumble across problem #2. But there are more problems that we need to focus on if we take reliability engineering seriously.

**PROBLEM #3** (*slow & expensive*) ... **INTERMEDIATE** AND **FINAL**  
DESIGN REVIEWS UNCOVER DESIGN PROBLEMS.



Problem #3 is something that many organizations have become ‘numb’ to ... because they don’t know any better. **BUILT-TEST-FIX** or **Test-Analyze-And-Fix (TAAF)** are so inefficient at making things reliable that they almost always result in lots more time and money being spent without ever realising their ‘reliability potential.’ These approaches involve rushing to create ‘functional’ prototypes and then ‘run’ them to see if they work.

What this means is that we only find problems when we review a design we have already created. We never try to prevent problems. This slows everything down A LOT. Finding out that you have used the wrong material, or a circuit board is too close to a hot exhaust manifold, or the brake

system doesn't work in wet conditions is now really hard to fix and takes a long time to do it. It often involves having to modify lots of other components or subassemblies, which of course introduce more potential weaknesses. And it doesn't have to be this way.

The next problem isn't all that bad ... and in fact is often a good thing.

#### **PROBLEM #4 ... INITIAL TESTING IDENTIFIES KEY DESIGN PROBLEMS (cheap)**



*... in initial testing*

Organizations that deliberately push their new product, system, or service to the edge of (or beyond) their **design specifications** are deliberately trying to create this 'problem' which identifies design weak points. 'Hunting **failures**' like this is a really good thing! This doesn't mean waiting until prototypes are built. But it might mean we create models, mock-ups or component level prototypes on the areas that are most concerning to us to quickly work out how we need to accommodate them in the design.

But the best 'problem' of all is not really a problem – because we never allow it to happen.

#### **PROBLEM #5 ... PROBLEMS IDENTIFIED BEFORE YOU INVEST TIME OR EFFORT INTO CREATING SOMETHING.**

*... the same as problems #1-4  
but we just found it before  
we designed it into our product*

This is where we use reliability engineering to harness the collective knowledge of your team, incorporate existing data, and otherwise learn from the information at hand to make sure that your **first design is a reliable design**. And it is simpler than you think.

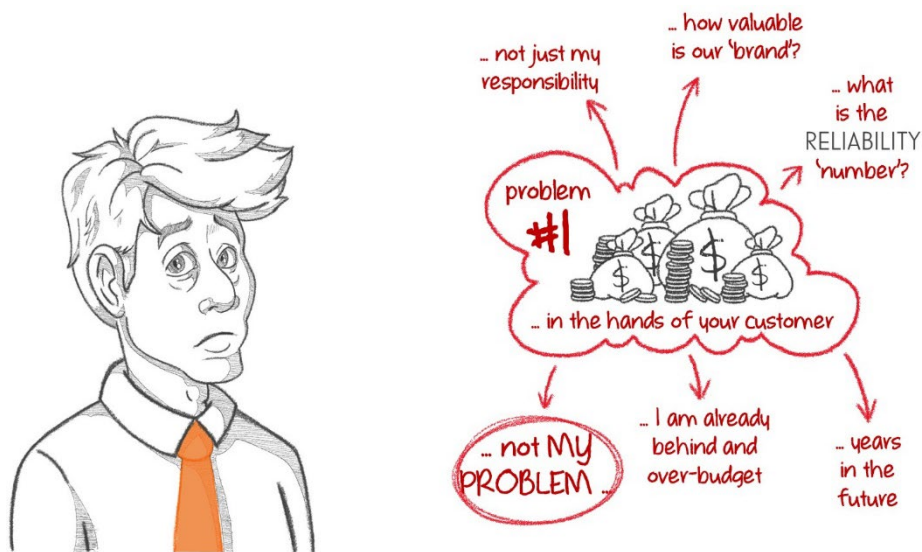
**Reliability engineering** prevents problems that cost a lot of money to fix or increase your **Time to Market (TTM)**. With your organization constantly 'fighting fires' there is not enough time and money to introduce fantastic design features to make your product 'better' than your competitors'.

Organizations who never link **reliability engineering** with preventing design problems that would otherwise be picked up in design reviews or final production prototype testing never understand how **reliability engineering** drives down production costs and schedule.

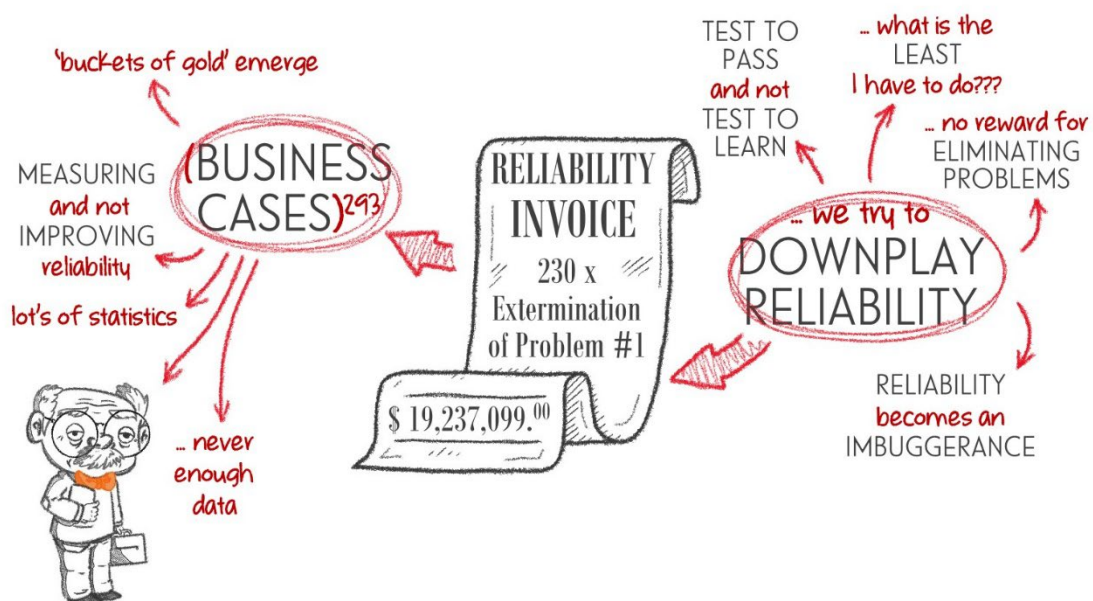
*... for RELIABILITY TO HAPPEN ... organizations need to BELIEVE that  
RELIABILITY ENGINEERING REDUCES production COSTS AND SCHEDULES ...*



If we instead only believe **reliability engineering** deals with problem #1 (and not anything else), young engineers like the one below have a hard time justifying to themselves that they should take **reliability** seriously.



And because there are no perceived benefits of investing your time into **reliability engineering**, there are only perceived costs.



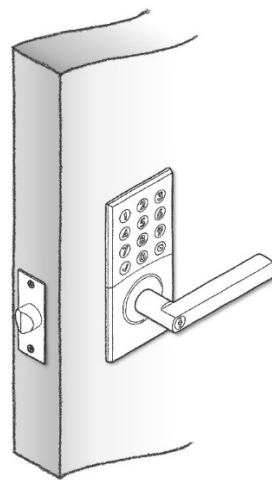
Business cases often get in the way of good **reliability engineering**. A good business case starts with having leaders who are keen to explore how **reliability engineering** can help them do business better. But then there are business cases demanded by leaders who see **reliability engineering** as an imbuggerance, and need every statistic, every eventuality, and every scenario rigorously interrogated before they make a decision. And by that time, it is too late. And of course, people are quickly going to stop making these business cases when it becomes too hard.

### Reason #3 – NO complex, expensive fixes

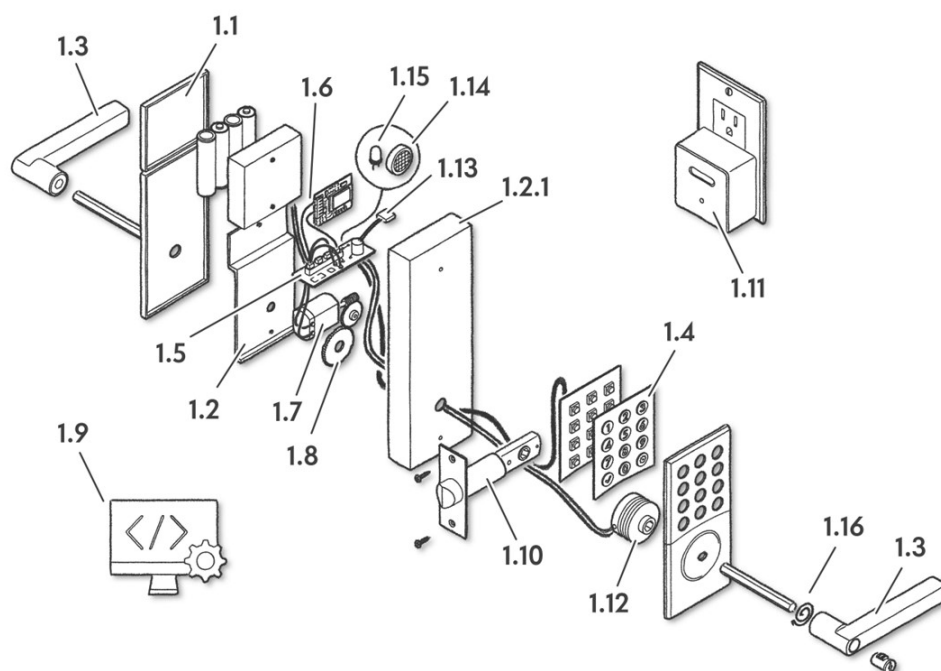
**Reliability engineering** only ever happens when we incorporate simple and effective design characteristics from the very start. Every problem (no matter how technologically complex your 'thing' is) has a range of solutions. The cheapest and easiest are ALWAYS those that can only be solved using a design characteristic BUILT INTO YOUR FIRST DESIGN.

*It is only when you are forced to redesign something without 'affecting' other components that fixes become complex and expensive.*

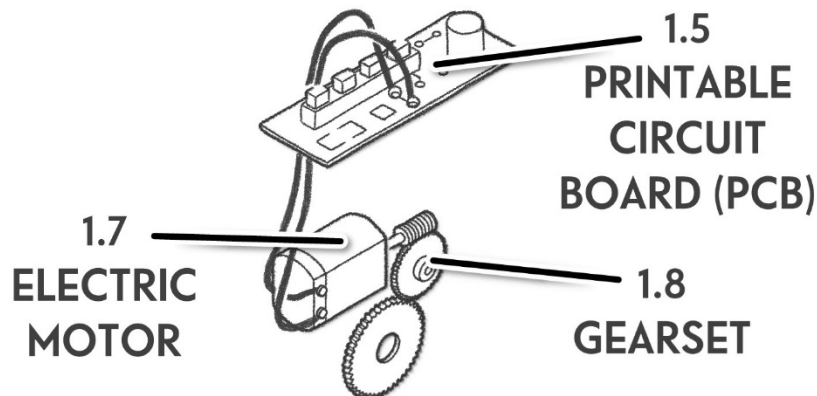
Think about a smart lock, which combines traditional 'mechanical' lock technology with contemporary 'smart' technology.



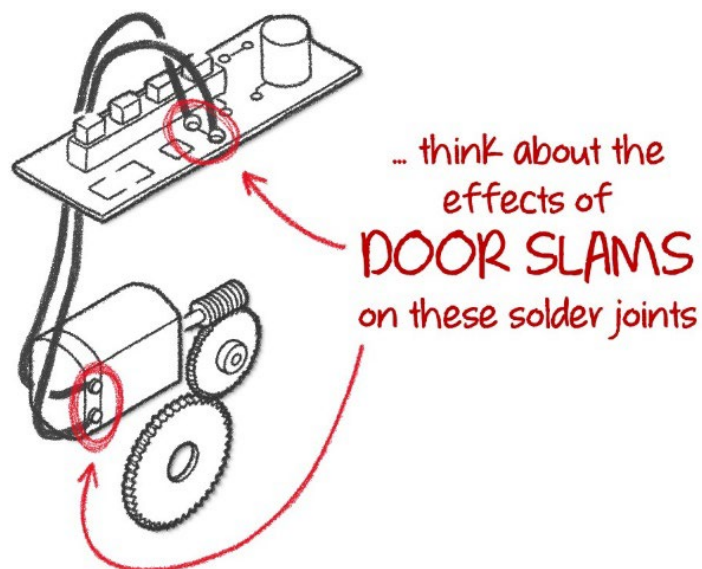
Even before we start designing something like a smart lock, we already know what many of the components will be and look like. A basic, preliminary configuration is illustrated below.



At the heart of the system is an electric motor controlled by electronic circuitry to transition the lock to or from a locked or unlocked state. Electric motor failure is 'critically bad.'



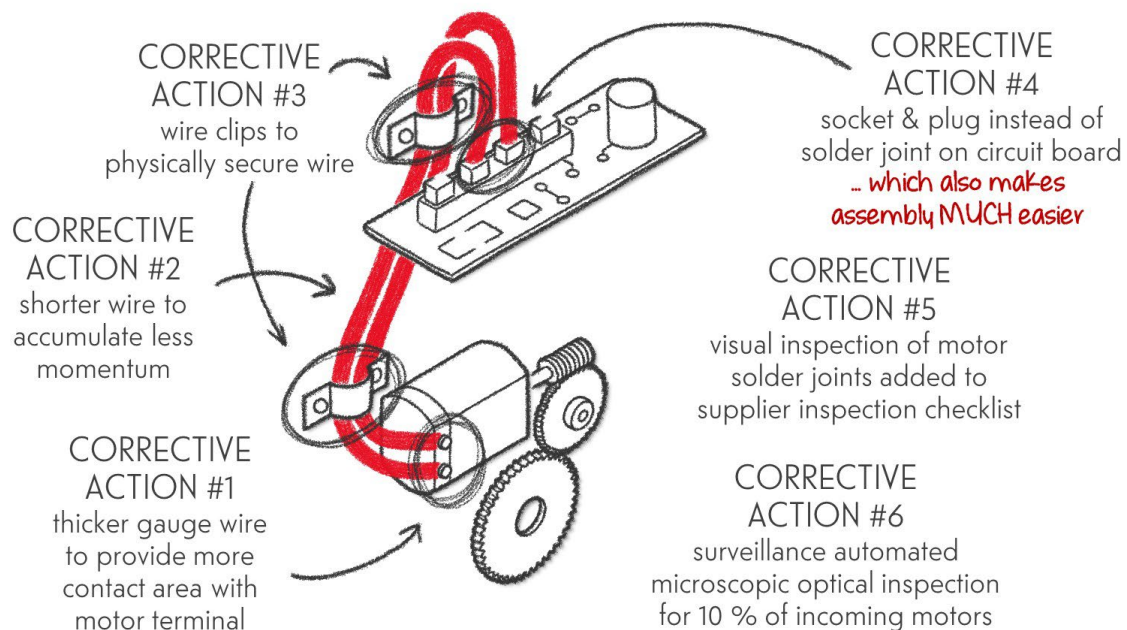
When we design something that is 'novel' or 'cool' we can often overlook the simplest of things. Like electric cables that connect the electric motor to the Printable Circuit Board (PCB). A **BUILD-TEST-FIX** approach means building a prototype that uses soldered cable that is not resilient to shock loads that are generated when a user or customer slams the door shut.



Perhaps qualification testing will uncover this critical design flaw ... as our lock needs to be able to withstand multiple shock forces when the door it is installed on is slammed (which a significant number of our future customers will no doubt do.) But many 'tests' are often based on the 'perfect' customer who treats our product carefully. Or at least, how 'we' think they should treat it. Unsecured cables in our smart lock only physically restrained with solder is a recipe for failure.

But instead of doing that, the smart lock's design team did 'smart' reliability engineering and perhaps conducted something like a **Failure Modes and Effects Analysis (FMEA)**. The key outputs are a set of **corrective actions** that address the design, manufacture, maintenance or anything else that can improve reliability.

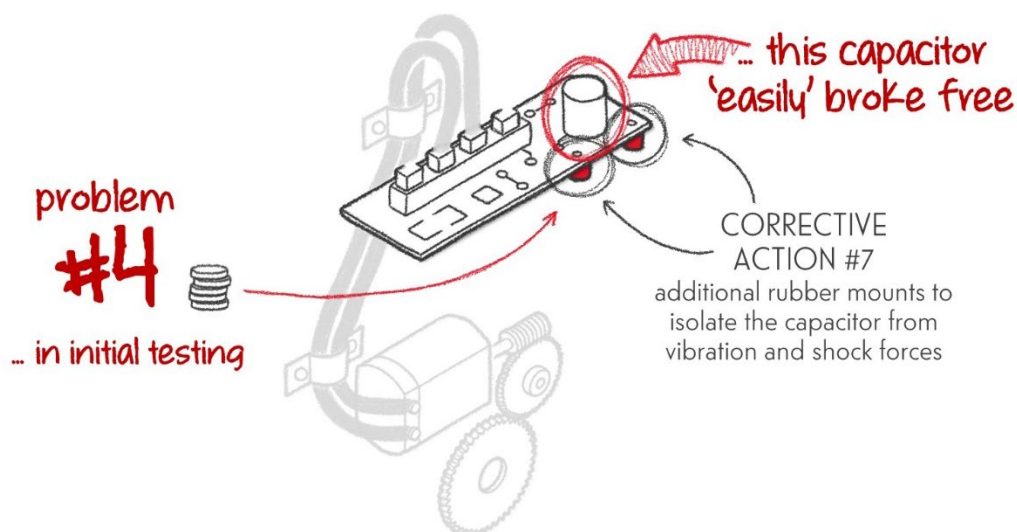
And look how simple and inexpensive these (real world) **corrective actions** can be.



These are all examples of 'problem #5' as we prevent many **failures** from every occurring by these fast, simple and (effectively) free **corrective actions**. And they are only ever fast, simple and (effectively) free if we embed them in our first design. Of course, there might be some additional material costs, but these costs will most likely be a negligible part of the recommended retail price.

But perhaps we still aren't as confident about the reliability of the PCB when the smart lock is subjected to door slams. PCB reliability can be difficult to understand, especially when it comes to vibration.

**Highly Accelerated Life Testing (HALT)** can be used to test a prototype PCB in minimal time and in minimal cost. This involves securing the PCB prototype to a vibrating table inside a chamber whose temperature is rapidly cycled from very high to very low to see how expansion and contraction might affect reliability. And any **failures HALT** uncovers are simply 'problem #4s'



This is something that a **BUILD-TEST-FIX** approach can't do. **This is how we make reliability happen.** It is virtually impossible (or at least very expensive) to put **reliability** into something that has already been designed or fielded. Some organization's hope that this is not true (to the extent that they perhaps believe in the supernatural ...) This means that **reliability engineers** are only called in for the final design review or qualification test when design and manufacturing processes are already frozen. *It is too late by this stage.*

We have already met our first enemy of reliability – the **ponderous professor** whose focuses on endless (and often frivolous) statistical inference and data analysis. The second enemy of reliability is the **infant manager**, and he gets in the way of making us think about any other problem besides problem #1.



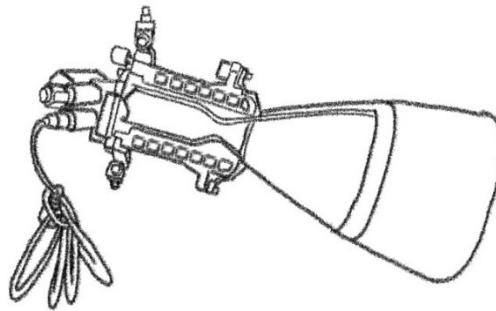
infant manager

The infant manager's most famous catchphrase is '**WTF!**' ... which stands for '**give me the WRONG THING FAST!**' He is focused on false metrics, meeting the first milestone as quickly as possible, cutting costs from the start and other things that provide the illusion of blindingly fast progress. But this illusion never lasts, because he or she pressures the team into not considering all those really simple and inexpensive design features that prevent the problems we talked about earlier.



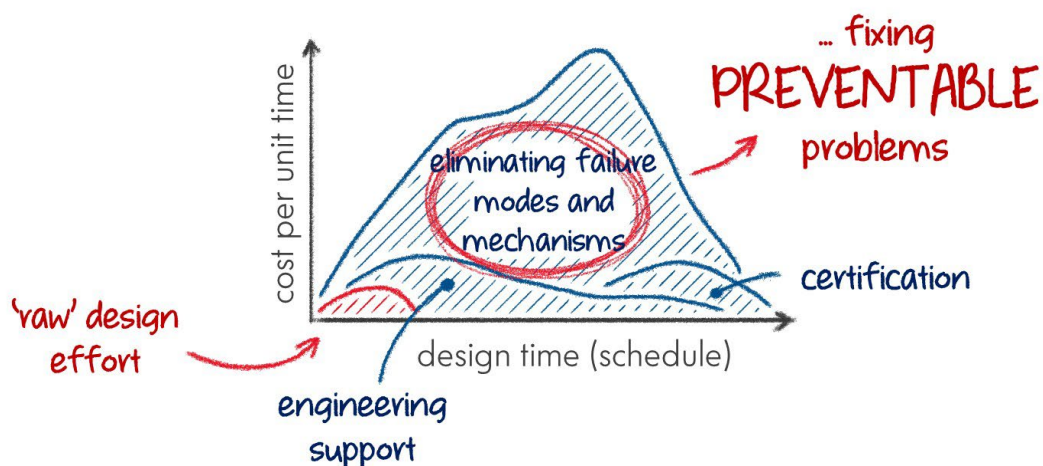
## Reason #4 – Saving (LOTS OF) time and money

Liquid fuel rocket engines involve many design challenges. High stresses and temperatures mean there are many ways they can fail. But this does not mean they can't be reliable.



'Traditional' design approaches are based on quickly BUILDing a prototype, TESTing it to find failures, and FIXing all the design issues behind those failures. The problem with this is that we wait until we construct something *before* we start thinking about making it reliable. So, the BUILD-TEST-FIX approach ends up taking more time and money.

We know how much time and money BUILD-TEST-FIX approaches take (for example) for liquid fuel rocket engines. <sup>1</sup>



This can be avoided if we *stop and think* about designing something reliable before we start designing. In short ...

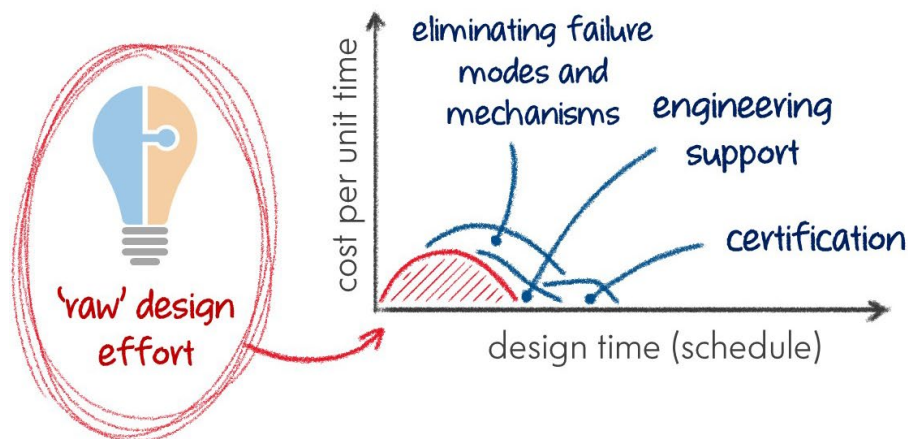
**... MAKING YOUR FIRST DESIGN A RELIABLE DESIGN.**

<sup>1</sup> Rajagopal, K., An Industry Perspective on the Role of Nondeterministic Technologies in Mechanical Design, Engineering Design Reliability Handbook (Niolaids, E., Ghiocel, D.M., and Singhal S Editors), CRC Press (2005)

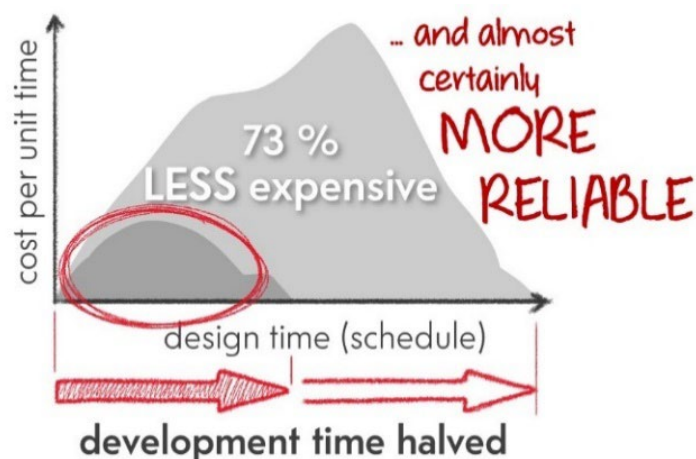
Reliability engineering involves thinking about failures from the very start, where everyone is encouraged to talk about what might go wrong in a constructive way. This allows us to prioritize the potential root causes of failure to a very small number that we call ...

... the 'VITAL FEW.'

And the illustration below shows how much time and money when we start using a RELIABILITY MINDSET.



Designing liquid fuel rocket engines with the RELIABILITY MINDSET means we might perhaps double our initial design effort when compared to the BUILD-TEST-FIX approach – but we then spend very little time and money on fixing preventable problems. If we start with a RELIABILITY MINDSET, the rewards are substantial. An actual study identified this benefit – which is illustrated below for the liquid fuel rocket engineering example.

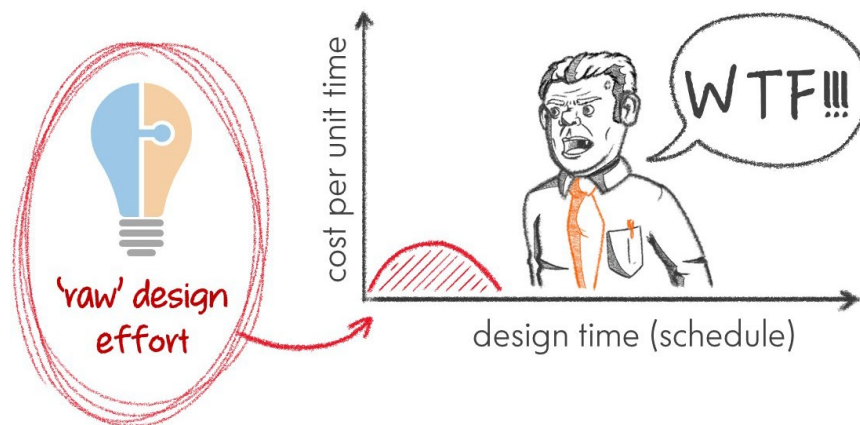


So what does this mean for you?

Firstly, there are huge opportunities for you and your organization to become extraordinarily successful. There is a tendency for us to focus on short term goals that involve more 'certainty.' Investing in a medium-term payoff with something as uncertain as reliability is daunting for many. Which means that your competitors are *likely* to be slower and less cost-effective than they could be.

And secondly, the opportunity for success might be really easy to grasp. All you have to do is harness the corporate knowledge your team already has. If they know how to make something that can work, they will also know the ways that it won't from work.

But remember, one of the reasons we don't see the reliability mindset is because our infant manager doesn't want to invest more time or money than he has to into that first, reliable design.



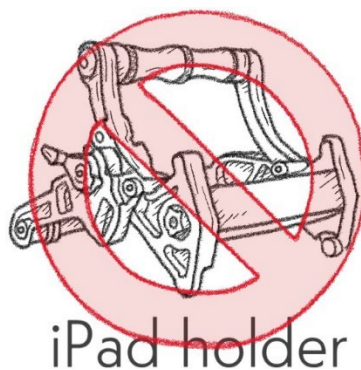
## Reason #5 – quickly solving the VITAL FEW

*Everything fails through a VITAL FEW ways.*

*Not the TRIVIAL THOUSANDS.*

Good **reliability engineering** is about quickly solving the VITAL FEW design problems. This gives us the biggest improvement in reliability with minimal investment. Which means **reliability engineering** is **not over-engineering** (like this iPad holder.)

Focusing on the TRIVIAL THOUSANDS provides an **illusion of reliability engineering**. All we are doing here is making the strong bits of our product, system or service ... stronger. **Over-engineering** means we waste money, time, resources, space, footprint and energy consumption that may (or may not) also involve improved reliability.



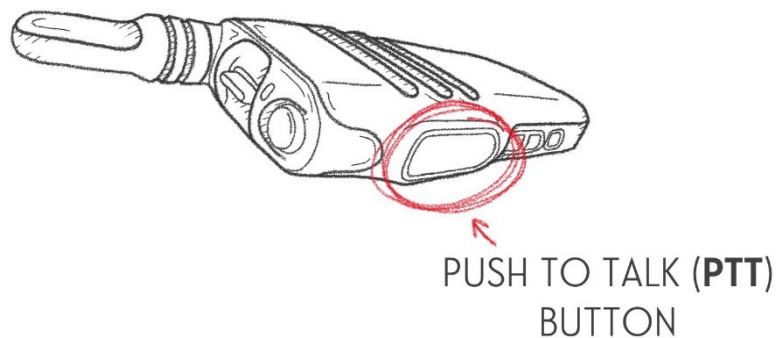
iPad holder

## Reason #6 – Making your thing better (than your competitors)

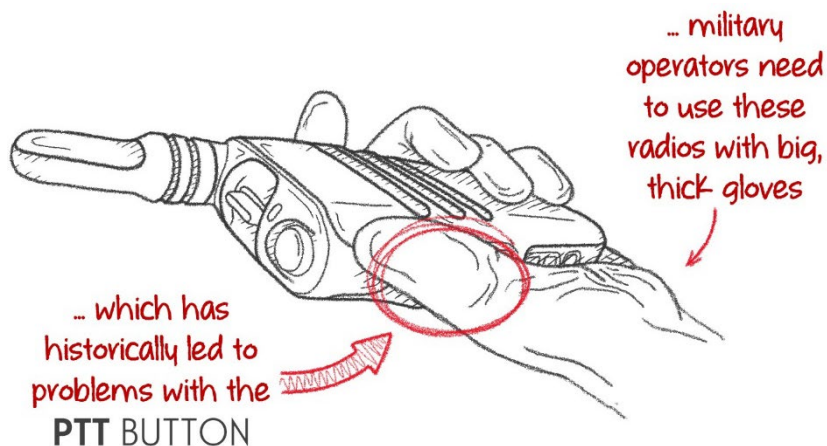
We have talked about how **reliability engineering** minimizes **problem #1 (failure** in the hands of our customer or user) through to problem #5 (something we design out of our first design). But there is another type of problem that reliability engineering helps us avoid.

**PROBLEM #X ... IT FAILS TO MEET YOUR USER'S OR CUSTOMER'S EXPECTATION** (*unrecoverable*).

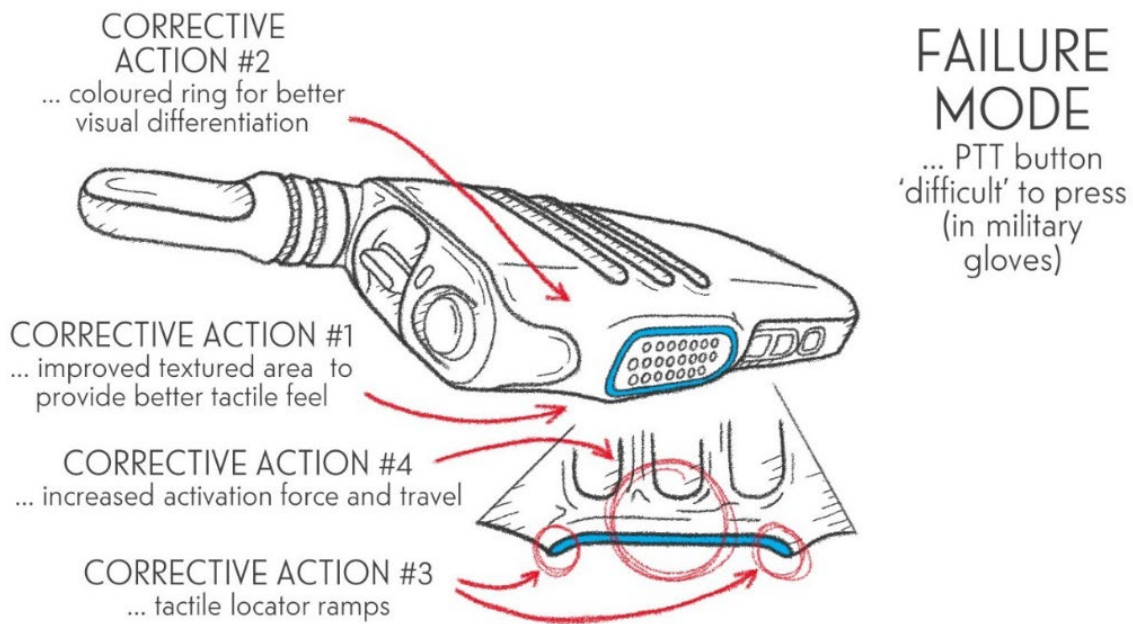
Good **reliability engineering** not only gives you more time to design better things, but helps you identify amazing design characteristics. All we need to do is look at failure from the perspective of the customer or user's expectations. Motorola used a **FMEA** to make sure its two-way radios have industry leading features. And they did this by focusing on the 'simplest' part of its radio, like the push to talk (PTT) button that allows users to transmit their voice across the network.



And they focused on a lot of different market segments, including military customers.



The **corrective actions** and design improvements from this **FMEA** are illustrated below.



Again, these **corrective actions** are fast, simple and (effectively) free if they are embedded into the first design.

## Reason #7 – Generating VALUE

*If reliability engineering isn't making you money, then you are doing it wrong.*

It is rare to find a 'detailed' case study of an organization that has successfully implemented a **reliability program**. We (as consumers) are often aware of which brands are more reliable than others. We know that Toyota manufactures some of the most **reliable** vehicles in the world ... and is also the world's largest auto manufacturer. But there is little more to talk about than this correlation (important as it may be).

Hewlett Packard however is a good example of what happens when **reliability** is taken seriously. It had received all manners of industry 'quality' and 'reliability' awards throughout the 1970s. They complied with all the relevant standards and actively marketed their devices as being superior in terms of reliability.

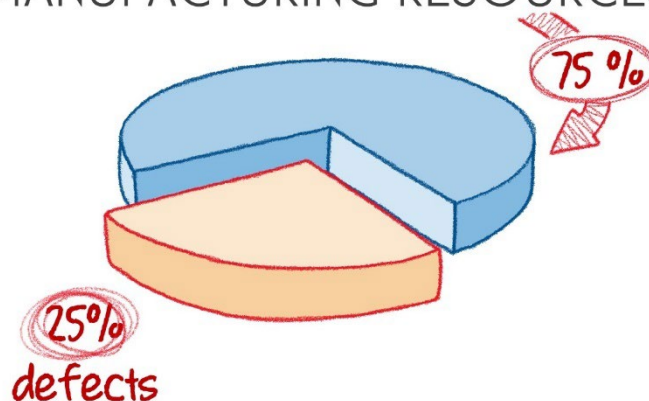


But then leadership began to ... lead. Specifically with the then CEO, John Young (below). He realized that even though Hewlett Packard was actively touting their **reliability** and **quality** performance, they didn't really know how well they were doing.



He launched an investigation, and (amongst other things) found that one quarter of the manufacturing budget was dedicated to fixing manufacturing or quality related problems *before* products were leaving the factory.

## MANUFACTURING RESOURCES

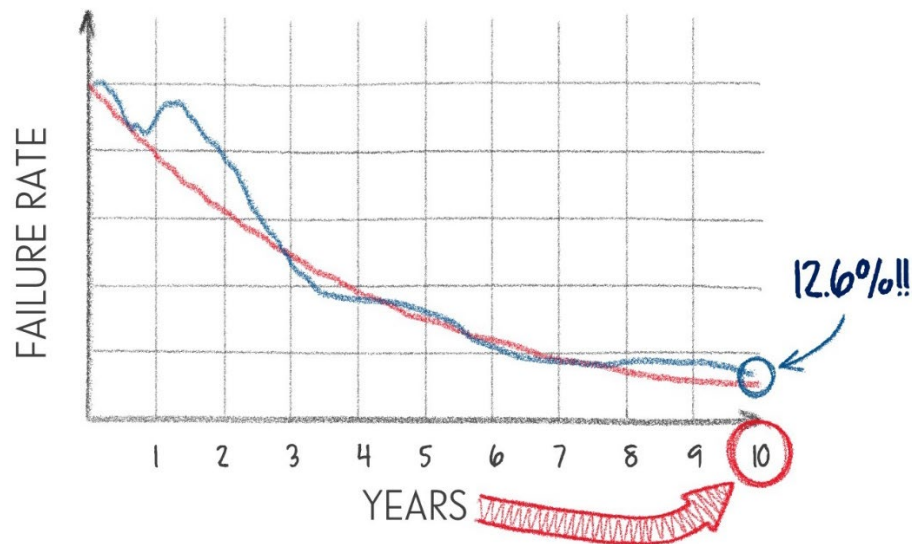


No one knew! And this phenomenon was repeated across many other budget streams where a lot of money was being lost to simply fix things that shouldn't need to be fixed. So he launched a project. It was ambitious – but also had total management commitment. This project had a name and a goal ...

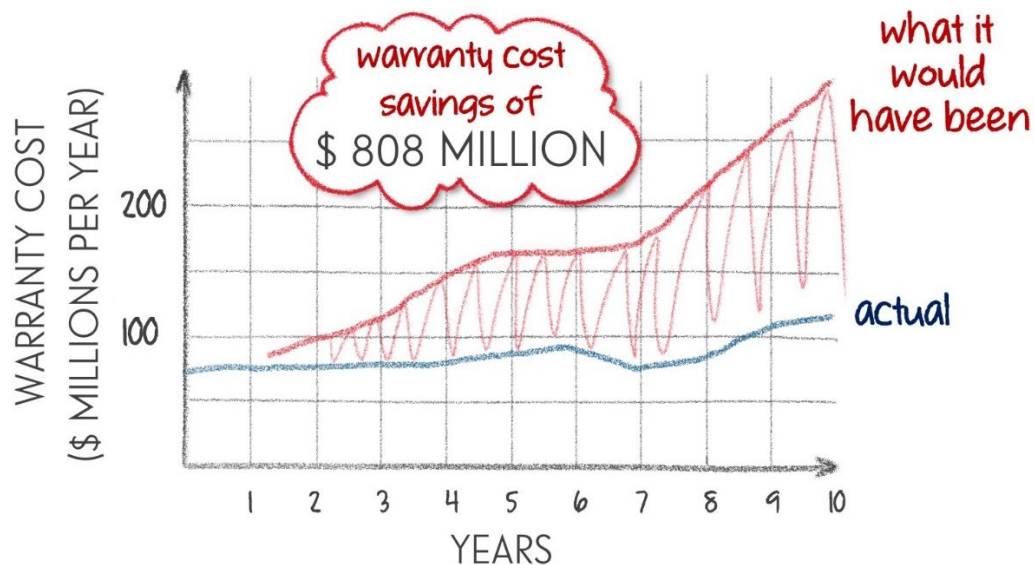
*The '10X Project' will reduce our failure rate (percentage of products experiencing warranty failures) by a factor of 10 over 10 years.*

The idea was to make a 'simple' goal that was easy to understand, knowing that not only would Hewlett Packard benefit from decreased warranty costs, but there would be savings and opportunities in other areas as well.

So how did Hewlett Packard go throughout the 1980s when they implement the '10X Project'? The didn't quite see a tenfold reduction in failure rate, but the project was still an amazing success. The red line on the chart below shows the theoretical and consistent reduction in failure rate needed to meet the goal of a tenfold reduction. The blue line represents what Hewlett Packard actually achieved.



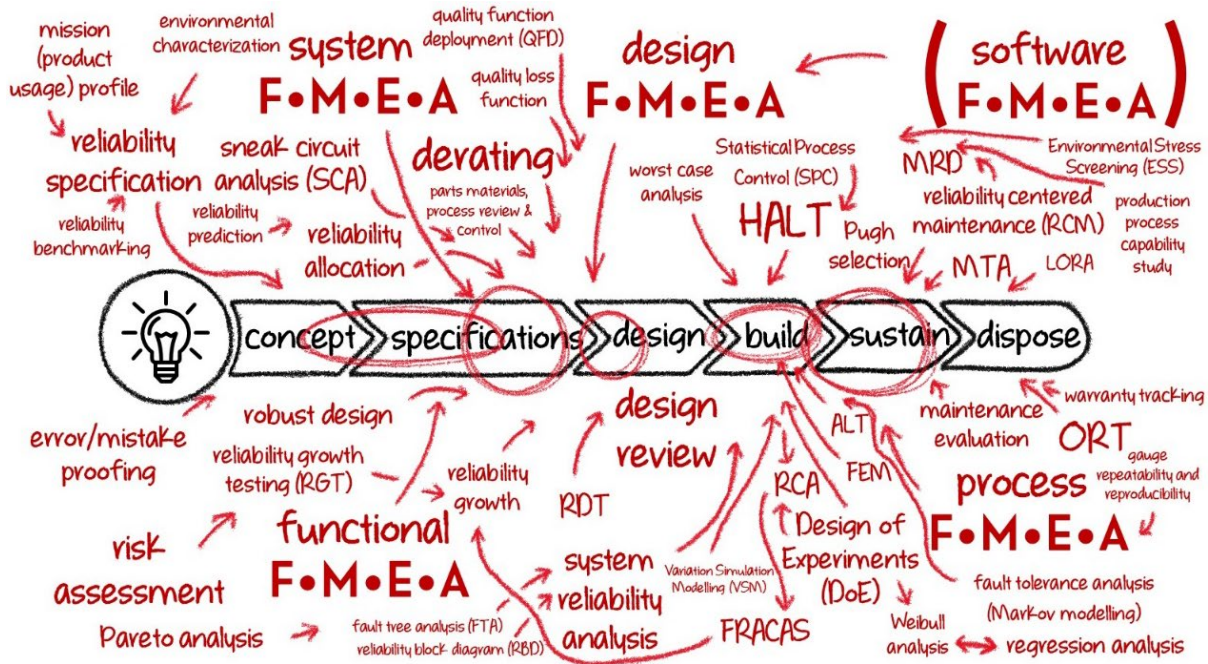
What did this mean? Well, the savings on warranty costs were substantial.



But it wasn't just warranty costs. Over **\$ 200 million was saved in warehousing costs** by year 3 because fewer spare parts were needed. Hewlett Packard formed strategic partnerships with suppliers and worked with them in a transparent way. One supplier was able to **decrease integrated circuit (IC) defects by a factor of 15 over 18 months**. Hewlett Packard didn't need to inspect them anymore. The **TTM** was reduced because **production time dropped 30 %**. The fewer problems they were experiencing allowed them to **drive down their recommended retail prices (RRPs) by 16 %**. Desktop computer **ongoing maintenance costs dropped by 35 %**. So not only were the products more reliability, the being designed and manufactured faster and cheaper. Value was everywhere.

## Reason #8 – No OVERWHELM

**Reliability** happens at the point of **decision**. **Decisions** occur across the product life cycle. And there are hundreds of **Design for Reliability (DfR)** activities you could use to help you make each decision.



... but this is NOT HELPFUL!

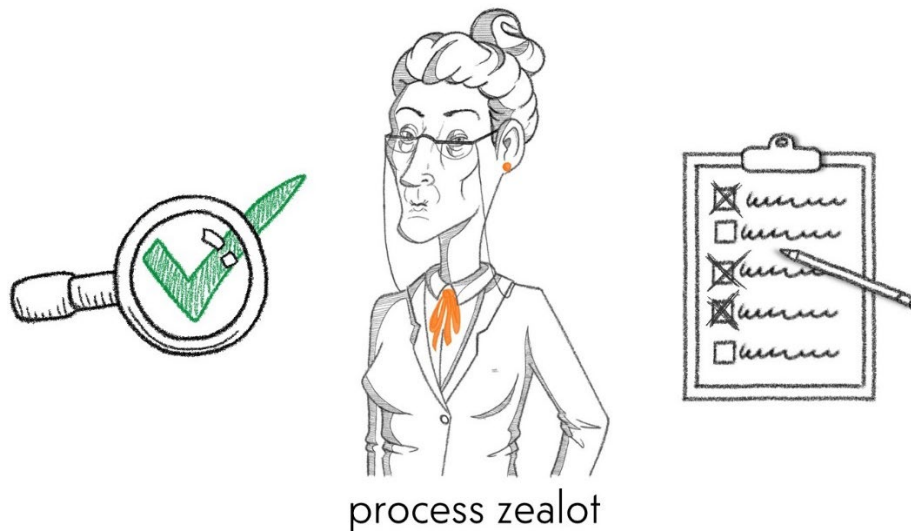
There are hundreds of tools in a hardware store ... but we don't find walking into a hardware store automatically overwhelming. Why? Because when we walk into a hardware store we usually know what we are trying to build, fix or achieve. So we know that we are looking for a hammer, or a saw, or a drill.

*Imaging walking into a hardware store not knowing what you are trying to build, fix or achieve BUT STILL buying the tools you will need to do 'it?'*

Many organizations have failed to make **reliability** happen because they keep everything complicated. They don't focus on their organization's *real* strategic goals, which means they don't create a simple strategy that allows everyone to understand their role.

... a *COMPLEX PROCESS* becomes the *PRODUCT* ...

This is where our third enemy of **reliability engineering** can get in the way – the **process zealot**. He or she doesn't care what you are trying to achieve. The only thing that matters are which activities (tools) you use.



This does not help. Trying to comply with every standard, textbook, custom, contractual clause et cetera 'steals all the oxygen' from **reliability** conversations. We don't spend time talking about improving **reliability**. All we do is talk about what milestones we have and haven't achieved.

### Reason #9 – happier people

The first reason good reliability engineering results in happier people is that we need to start with eliminating the enemies of reliability. And this always makes people happier.



PROCESS ZEALOT



INFANT MANAGER



PONDEROUS PROFESSOR

**Good reliability engineering is engaging and rewarding.** It is all about selecting the VITAL FEW (effective) reliability tools to drive high quality and reliability based on the circumstances of the program being supported, based on reliability plans (that work) that identify and deploy the right tools for the program.



## Reason #10 – your REPUTATION

Good **reliability engineering** means you create more value for your organization. So you become more valuable. You experience fewer problems (let alone failures). Your products, systems or services are just better. And when you are valuable, you will inevitably get rewarded for it.



And it comes down to good **reliability engineering**:

- 1. reliability must be DESIGNED INTO PRODUCTS AND PROCESSES.*
- 2. knowing how to calculate reliability is important - but KNOWING HOW TO ACHIEVE RELIABILITY IS MORE IMPORTANT.*
- 3. reliability practices must be INTEGRATED into overall product development and design processes.*

If you don't do things, then essentially you are going to hope that reliability will 'happen' sometime down the schedule. Do you believe in the reliability fairy who can 'sprinkle' reliability over an already designed and manufactured prototype?



## RELIABILITY FAIRY?

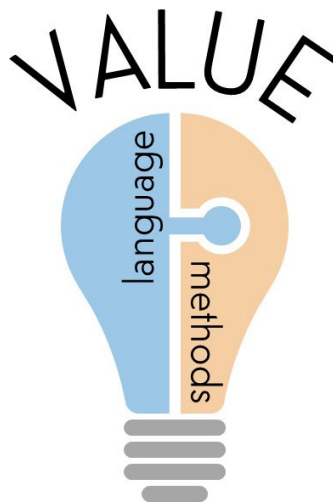


## EXERCISE

What is your 'FAVOURITE REASON' for doing reliability?

It starts with the RELIABILITY MINDSET

Which means that every design, manufacturing, operation and maintenance decision needs to be influenced by reliability. That sounds overwhelming! But reliability only happens we make sure it *isn't* overwhelming. The reliability mindset always starts with VALUE.



Understanding the relationship between value and reliability means we know how each reliability activity will (1) drive down production costs, (2) accelerate development, (3) result in products with more functionality, (4) reduce over-engineering and (5) ... be more reliable!

The second part of the reliability mindset is the LANGUAGE. What does 'failure mode' mean? What is the difference between reliability and the Mean Time Between Failure (MTBF)? What is 'infant mortality'? ... and is that different to 'wear-in'? Making sure everyone knows how *your organization* describes reliability metrics and concepts is absolutely essential.