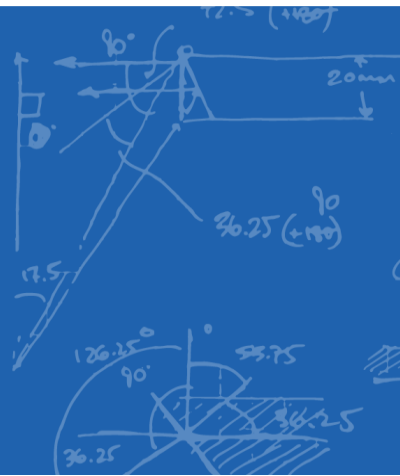




WHITE PAPER | JULY 2021

A PRIMER: Developing a Product Requirements Document



The Market Leader in Mechatronics and Detailed Engineering Design Services

simplicitypd.com

INTRODUCTION

In this whitepaper we will discuss the importance of documenting product requirements prior to starting product design work. We will examine best practices around developing requirements documents, discuss translating user needs into *actionable* product requirements, and provide guidance on appropriate requirements language with examples of both well and poorly articulated requirements. In addition, we will provide an editable Product Requirements Document Template that you can use to document requirements for your next project.

THE IMPORTANCE OF A PRD

A Product Requirements Document (PRD), sometimes referred to as “product requirement specification”, is a set of product level requirements imposed upon a design which describe “WHAT” the product must do or qualities it must have, while carefully avoiding specific design solutions.

The PRD is the **CORNERSTONE** to any successful product development project and should be created *early* in the product development lifecycle (e.g., during the [Requirements and Planning](#) phase of a project). Jumping into detail design work without a good set of requirements is not recommended. The PRD will serve as a type of contract between engineering and the product owner or client/customer, communicating a clear and agreed upon understanding of the product requirements necessary to develop the right product.

Good requirements documentation saves both time and money by eliminating iterations of the design due to misunderstandings that arise from incomplete, ambiguous, or conflicting requirements. If budget and schedule are important to your project, it is well worth the effort to take the time upfront to define a good set of comprehensive requirements.

3 KEY STEPS TO DEVELOPING A GOOD PRODUCT REQUIREMENTS DOCUMENT

1. Define User Needs
2. Translate User Needs into Product Requirements
3. Review Requirements for Adequacy

1. DEFINE USER NEEDS

Before you can start creating a good set of product requirements, you must first develop a clear understanding of the intended use of the product, including the needs of the user (and patient if a medical device), as well as the target geographical markets for which the product will be sold due to its impact on regulatory, languages, and social norms.

Intended use focuses on what the product needs to do (i.e., what its purpose is), who will use it (what limitations or skills may be required), how it needs to be used and for how long, under what environmental conditions it needs to be used, stored, transported, etc.

Typically, a separate user needs document (also referred to as a “user” or “marketing” requirements document) precedes a product requirements document. This is a separate topic and will not be fully discussed in this paper. However, it is also important to emphasize that user needs should be based upon input from actual users of the product and not just on market research and brainstorming. It is critical to **get your user needs right**. User needs form the basis of validating that you have **designed the right product** for its intended use.

2. TRANSLATE USER NEEDS INTO PRODUCT REQUIREMENTS

Once the user needs are defined, the next step is translating those needs into product requirements. Creating a product requirements document is not an activity of a single individual; it requires input from all key stakeholders to ensure all important aspects of the product’s design requirements have been identified. Typically, the key stakeholders are going to include members of a cross-functional team, such as engineering, marketing/sales, clinical (if medical device), quality/regulatory, service and manufacturing, along with the user and client or customer.

A comprehensive Product Requirements Document should consider the categories listed below.

- Physical Characteristics
- Function
- Performance
- Interfaces
- Environmental Conditions
- Reliability
- Maintenance and Service
- Manufacturing
- Packaging and Shipping
- Safety and Regulatory Requirements and Standards
- Labeling

The process of developing these requirements is best described as asking a series of questions regarding the user needs within the framework of these categories. Inputs can be documented using a Product Requirements Document Template. See examples of sample questions that follow.

Physical Characteristics: What are the physical attributes and characteristics needed based upon user preference or how it will be used?

- Consider size and weight limitations, color, material compatibility, any other physical requirements of the product.

Function: What features and functions must the product have or do to serve the needs of user?

- Consider power, connectivity, and other operational capabilities of the product and processing of inputs and the resultant outputs.
 - Any special power requirements?
 - Battery/low power?
 - Battery life requirements?
 - Battery charging?
 - What is the list of motors, heaters, actuators, and sensors that must be electrically controlled?
 - What are the I/O interface? USB, serial, Ethernet, Wi-Fi, Bluetooth, I2C, SPI, etc.?
 - What logical states does it go through?
 - What triggers transition between states?
 - How is/are the operation(s) started and stopped?
 - What actions are needed upon completion?
 - Sensor interfaces?
 - Actuator interfaces?
 - Signal processing requirements?
 - Control algorithm requirements?
 - What data logging is required?
 - How is this configured?
 - How is it started/stopped?
 - Where is the data stored?
 - How are logs managed? (bounded space? bounded timeframe?)

Performance: How well do the product functions need to perform their tasks?

- Consider speed, accuracy, noise, force limits, flow rate or other characteristics your product must have to perform successfully or effectively.

External Interfaces: What else does the product need to interface with and what is required for that to be successful?

- Consider products, peripherals, accessories, consumables, different HW or SW versions, ports, connectors, power, etc.
 - I/O with external system?
 - Does the product interact with a host PC, mobile device or networked computer?
 - Other SW? (databases, network, etc.)
 - Other devices? (vision systems, motor controllers, sensors, data acquisition systems, etc.)
 - What operating system(s)?
 - Is a SW installer required?

User Interfaces: How does the user need to interact with the product, and how does the design support safe and effective use of the product?

- Consider physical aspects, as well as languages, character sets, auditory, tactile displays, visual, etc.
- What is on the user interface?
 - Controls?
 - Status information?
 - What messages & data displays are needed?
 - Access controls/permissions?

Environmental Conditions: In what environmental conditions will the product be used or stored?

- Consider temperature, humidity, exposure to liquids, altitude, salt spray, chemicals, electromagnetic fields, vibration, shock, impact, etc.

Reliability*: How long does the product need to perform its intended function adequately and safely?

- Consider probability of success, duration (time, cycles, etc.), environment, required maintenance, Mean Time Between Failures (MTBF) or Mean Time to Failure (MTTF)
- Consider both system and sub-system components.

**For more on Reliability see Appendix on page 13*

Maintenance and Service: Will the product require maintenance or service?

- Consider what parts need maintenance or service, frequency, accessibility, tools, level of skill, location, etc.
- Consider the costs of service calls vs. cost of reliability testing

Manufacturing and Test: Will the product perform any manufacturing functions or tests?

- Special production line mode required?
 - Is manufacturing support SW package needed?
 - Acceptance testing?
 - Factory calibrations?
 - How/when will they be performed?
 - Where should the data be stored? (Embedded FW, SW)
 - Additional capabilities of that mode?
- Special engineering mode required?
 - Additional capabilities of that mode?

Packaging and Shipping: How does the product need to be shipped?

- Consider contents of shipping package, sterility requirements, shipping standards (e.g., ISTA, ASTM D4169, etc.) that need to be met to protect the integrity of the product and packaging, whether it will be by ground or shipped by air, single, consolidated, or palletized shipments, environmental conditions (temperature, vibration, drop, etc.).
- Consider whether the product contains any dangerous goods (e.g., Lithium batteries, magnets, hazardous chemicals, etc.) as defined by transportation agencies such as the International Air Transport Association (IATA) or US Department of Transportation (DOT). Shipment of dangerous goods may impact your packaging design, method of shipment, required testing and certification, and labeling.

Safety, regulatory, and standards: What are the potential safety and environmental hazards, and how will the design mitigate these? What regulations and standards is the product required to comply with?

- Consider the type of product (medical, electrical, etc.) and where it will be sold, domestic or foreign, including any local state requirements. Determine what regulations (FDA, EPA, OSHA, REACH, etc.) you are required to comply with and what directives (Medical Device Directive, RoHS, WEEE, etc.) or standards (IEC 60601-1, IEC 62304, ISO 10993-1, etc.) you choose to comply with.
- Consider what risk control measures (mitigations) are required to address any safety or environmental hazards identified during your risk analysis activities. A risk analysis provides critical input to a product requirements document. All identified risk control measures should be linked to a product requirement. Identifying risk control measures often drive changes to your product requirements document by clarify existing requirements or adding new requirements.

Labeling: What product labels and user documentation is needed?

- Consider the information that needs to be conveyed, including regulatory and standard requirements for labeling, warnings, certifications (CE Marking, UL, etc.), number of product labels, placement, and durability.
- Consider information the user needs to properly install, operate, maintain/service the device - Instructions for Use (User Manual), Service Manual, Installation Guide, training materials, etc.
- Consider whether any of your labeling will be in electronic format or contained in the device software.

3. REVIEW REQUIREMENTS

Review product requirements for adequacy. This review team should be multi-disciplinary, most likely the same key stakeholders previously identified. To determine adequacy of a requirement, it is best to consider the following guide for writing good requirements.

Simple requirement formula:

The (noun “object”) shall (verb “action”) (criteria/requirement)

Examples:

- The *device* shall *weigh less than 20 lbs.*
- The *system* shall *provide six degrees of freedom control.*
- The *device* shall *operate for 12 hours on a fully charged battery.*

11 Key Characteristics of Good Requirements

Well defined requirements share key characteristics:

1. Unambiguous
2. Clear
3. Verifiable (Testable)
4. Atomic (a standalone statement)
5. Correct Use of Terms (“shall” and “should”)
6. Implementation-free (they define the “what” not the “how”)
7. Correct
8. Consistent (non-conflicting)
9. Complete
10. Necessary
11. Traceable

1. **Unambiguous** – there should be only one way to interpret a requirement. Requirements should not be vague or too general as they can result in multiple and incorrect interpretations.

Eliminate the use of inherently weak or vague words such as “user-friendly”, “efficient”, “easy”, “effective”, “reliable”, “few”, “timely”, “support”, “enhance”, “support” “but not limited to”. Due to their vagueness, use of these words leads to unverifiable requirements.

Bad		Good
The device shall be <i>easy</i> to lift.	➔	The device shall weigh less than 10 pounds.
Where possible the operational noise of the device should be <i>minimized</i> to not cause harm.	➔	The device shall operate at an acoustic noise level less than 80 dBA as measured over 24 hours of continuous use at 3 feet away from the source of the noise, according with IEC 60601-1.

In the “bad” example above, the words “easy” and “minimize” are vague and there is no clear way to verify compliance.

2. **Clear** – a requirement should be concise, simple, precise, and understandable.

- Avoid using unnecessary verbiage or information
- Expresses objective facts, not subjective opinions
- Be grammatically correct
- Avoid or limit the use of negatives (e.g., “shall not”)
- Avoid industry specific or company jargon, acronyms that are undefined
- Use the “active” voice so it is clear “who” is responsible and “what” must be done

Bad		Good
The packaging should be made of stiff materials so that the device is not damaged if it is shipped and happens to get dropped.	➔	The packaged device shall withstand ISTA 3A Series Transportation Test without loss of safety or essential function per IEC 60601-1.

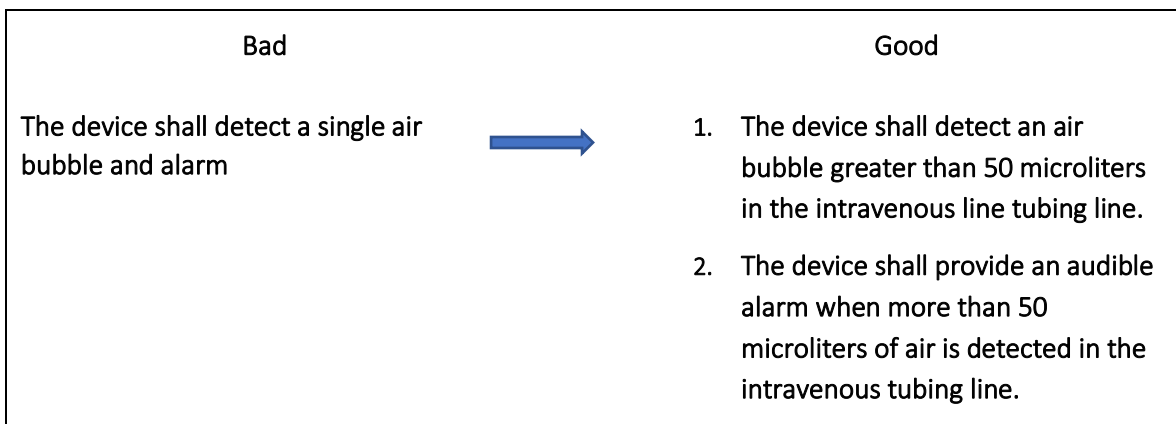
3. **Verifiable (Testable)** - all requirements must be verifiable by an objective method of analysis, inspection, or testing. For a requirement to be verifiable it must also be clear and unambiguous. It is important that the requirement contain criteria for verification, preferably a measurable quantity. See preceding examples.

COST OF TESTING

All requirements using the “shall” terminology *must* be verified and many, if not most, will be formally tested to prove that the design satisfies the requirement. Put another way, if a test is considered too expensive to be run, then the requirement that the test is intended to satisfy must not really be a requirement.

Testing is a non-trivial cost. It requires generation of a test plan, typically some sort of specialized test equipment or software, labor to run the tests, labor to analyze the test results, and often expensive prototype units to test. Depending upon the level of design challenge as driven by the requirements, it is quite common for some portion of test to fail, requiring design iteration and retest.

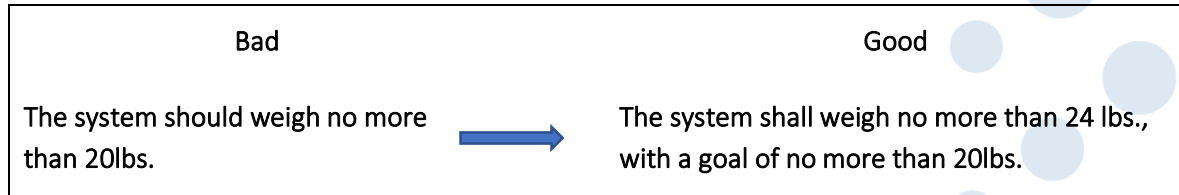
4. **Atomic** –a single thought, a standalone statement, as opposed to multiple requirements in a single statement.



The “bad” example above the requirement has combined 2 different requirements; each of these requirements need to be written as a single traceable requirement along with specifying criteria that enables verification.

5. **Correct Use of Terms** - uses “shall” or “should” properly and consistently; “shall” indicates a requirement, which is mandatory and requires verification, while “should” is a guideline, goal, or recommendation and does not require verification.

Requirements should be written using “Shall” statements. Sometimes, however, there is a need by the stakeholder to convey a goal or target for the designers in hopes of making every effort to hit that goal. Best example is in threshold or limit setting, where you first state the requirement that is to be verified, followed by the goal.



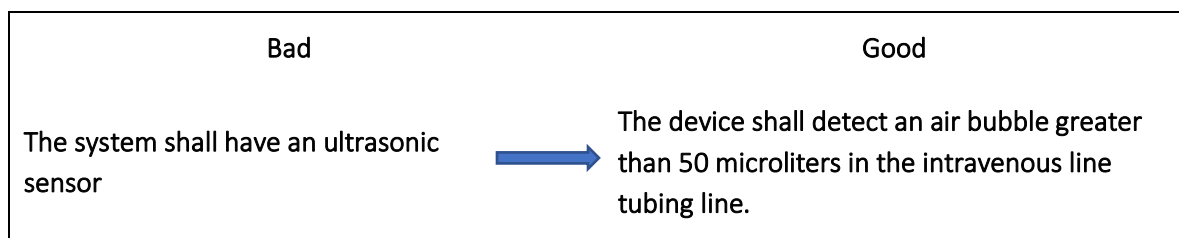
6. Implementation-free - defines the “what” not the “how”.

System requirements in a product requirements document define “what” the user and the other various stakeholders of the product need written in terms of “what” the product must do or the qualities it must have. Often, however, the requirements are mistakenly written in terms of design and instead define “how” the design is to be implemented. Design specifications define “how” the requirements will be fulfilled.

There can be many ways to implement a solution to a problem; a requirements document should not limit the designer’s ability to define the best implementation.

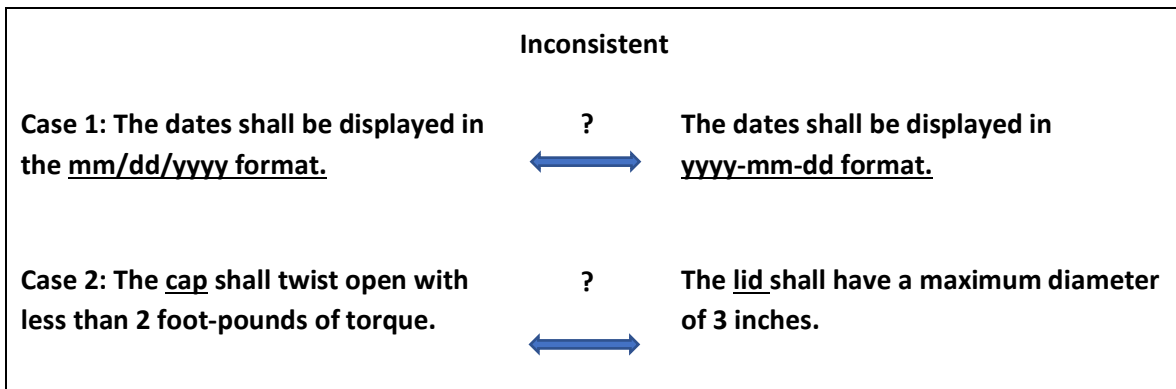
A requirements document that defines the “how” instead of the “what” not only takes away the designer’s freedom to determine the best solution, but also risks missing some critical requirements that can lead to a solution that does not adequately fulfill the needs of the user, customer, and other key stakeholders.

To avoid the implementation trap – it is best to ask “why” you need this requirement. If it is a specification, the question should lead you back to the real requirement (the need about “what” the product must do).



In the “bad” example above, the requirement is providing design detail on how to implement a solution. You need to ask, “why do you I need an ultrasonic sensor?”. Asking “why” will lead you to the real requirement as shown in the “good” example, which is to be able to detect air bubbles > 50 microliters.

7. **Correct** - accurately captures the expectations by the user, customer or client, and other key stakeholders. Assumptions and facts need to be checked by reviewing with all the key stakeholders to ensure they are correct.
- Are you referencing the correct standards that are applicable to your device?
 - Have you properly applied the requirements of the standards to your device?
 - Have you interpreted the user need correctly and considered acceptable resources for human factors applications?
8. **Consistent** – does not conflict with any other requirement and is consistent in its use of terms when describing the same object or characteristic in two or more requirements. A requirement is defined as conflicting when the implementation of one requirement prohibits the implementation of another requirement – essentially the two cannot realistically exist at the same time.



In case 1 above, the 1st and 2nd requirements conflict - we cannot have 2 different formats for the same thing.

In case 2 above, the 1st and 2nd requirements are inconsistently using terms to describe the same object. One describes the object as a “cap”, while the other describes it as a “lid”.

CONFLICTING REQUIREMENTS

When discussing requirements, one can be misled by the term “requirement”. While the full team from marketing through to test engineering intends the requirements to be truly required, the fact of the matter is that the initial PRD often represents a wish list. Once the design process has begun, starting with architecture, the process of learning about the product’s embodiment begins. As that process proceeds, it is not uncommon to find that some requirements are in conflict. The more challenging the requirements are to meet, the more likely this is to occur. Sometimes these conflicts can be resolved with redesign, but in some cases, they will simply represent a physical impossibility that was not evident until enough work was performed to discover it.

A simple example of conflicting requirements often arises in wearable electronics. It is always desirable that a wearable product to be feature rich, small, light and have long battery life. Batteries have their technological limits for energy density. Unfortunately, features tend to consume power which requires a larger and heavier battery. Power needs and viable battery size/weight are physical limits to technology that will impact the design.

Once a set of conflicting requirements is identified, it is up to the full team to decide which requirement(s) to relax to achieve a feasible design. The design engineering team should take the responsibility of defining a set of tradeoffs between the conflicting requirements and provide these to the business and marketing teams so that they can make an informed decision. In the example case of the wearable, the engineering team should be able to provide tradeoffs between battery size and battery life for different sets of features. In that manner, the business and marketing teams can optimize the relaxation of requirements to have the least impact on product value.

- 9. Complete** –a single requirement fully stated in one place with no missing information that could lead to misinterpretation or lack of fulfillment of the need. Sometimes this occurs because we do not define the applicable conditions for which a requirement applies.

Bad	Good
That device shall operate at less than 80 dBA.	The device shall operate at an acoustic noise level less than 80 dBA as measured over 24 hours of continuous use at 3 feet away from the source of the noise, according with IEC 60601-1.

In the “bad” example above there is missing information about the conditions for which the noise level applies.

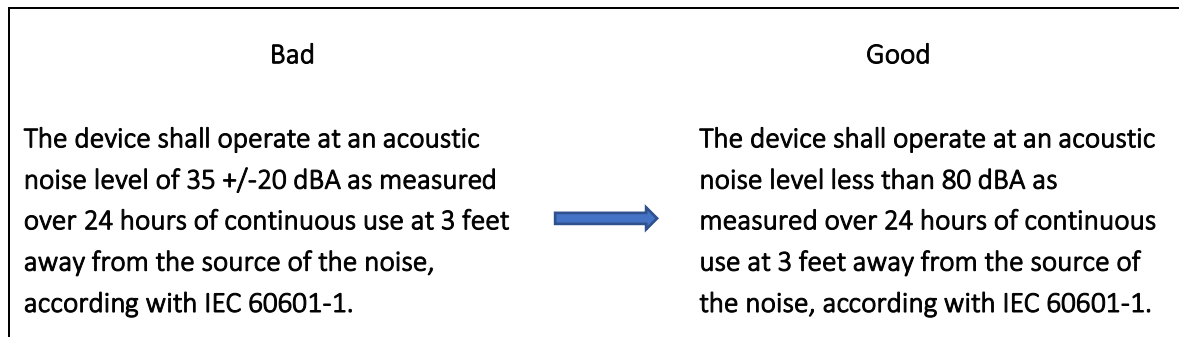
Completeness also applies to the product requirements document itself. Have we specified all applicable requirements?

To ensure all necessary requirements have been identified, it helps to follow an outline, or a list of types of requirements (physical, functional, performance, regulatory, etc.).

Conducting requirement reviews with key stakeholders through-out the product development process will be important in identifying missing requirements. New requirements may be identified during development because of learnings from prototype builds, engineering and usability testing, or ongoing risk analysis.

10. Necessary – is needed to define what the product must do or the qualities it must have for it to fulfill the needs of the user and other key stakeholders.

It is easy to get carried away and add features or functionality we think the user may want, or we over specify and make a requirement more stringent than is necessary. In both cases, we create additional, unnecessary, and costly work and risk forcing the design down a path that may not be optimal.



In the “bad” example above we have stated a noise level that is not required per the customer. The customer has only asked that the operation noise not cause harm. In addition, we have added a lower bound that makes it both more stringent and unnecessary.

11. Traceable –each requirement must be uniquely identified.

Each requirement in a product requirements document must have a way of being uniquely identified and referenced so they can be traced throughout the product development cycle. Once assigned, the requirement ID should not change. Consistency of requirement identifier supports clear communications and is more important than contiguous identifier lists.

CONCLUSION

Developing a complete and well documented set of product requirements provides clarity and efficiencies to your team's product design process. Documented requirements help confirm that the needs of the users and key stakeholders have been addressed, as well as differentiate between "must haves" and "wish list" items which may or may not be accommodated in the final design. Using clear and appropriate language to articulate requirements is equally important to ensuring clear communication to guide the design process.

It is important to remember that requirements will evolve during the product development process. There is a point of diminishing returns on investment in requirements at the start of the project. It is important to address as many requirements as possible up front, but it is also important to balance that with project schedule needs. Getting a requirement wrong and changing it late in the schedule is costly, but iterating requirements until they are perfect is also costly. Like many aspects of product development, it is a balancing act.

We trust you will find these best practices useful in helping guide your next design project and encourage you to leverage the companion **Product Requirements Document Template** we've developed to help you through that process.

APPENDIX – RELIABILITY

Reliability is a concern for every product. Everyone would like their design to be 100% reliable for the life of the product. Like any feature of a product, reliability comes at a cost, one that can be quite high. As product complexity increases, the number of failure points also increases thus naturally decreasing reliability. When discussing reliability, there are two concepts that need to be considered:

1. The impact on product cost and maintenance cost for high reliability.
2. The cost of testing required to verify that a product meets its reliability requirement

Product and Maintenance Costs

Three common strategies to provide high reliability products are (1) to use high quality, long life components, (2) implementation of redundant components, and (3) to schedule regular maintenance to fix problems before a product fails. A common example is commercial aircraft. Aircraft are highly complex products. They achieve extremely high reliability using all three strategies:

- Aircraft components are designed to be high quality and robust. The term “aerospace grade” is synonymous with both quality and expense.
- Commercial airliners have multiple engines. Engines are very expensive, but airplanes can continue to fly after an engine failure for the added reliability.
- FAA procedures mandate a pre-flight check by the pilot. Inspection before every use, as well as rigorous maintenance are used to head off any issues before a failure. Both require attention from expensive flight and ground crew personnel.

Reliability Testing *(Also see: Sidebar – Cost of Testing on page 7)*

The cost of prototype units required for reliability testing can be quite high. The prototype units under test are usually very expensive, and many units are required to provide high reliability at a high confidence level. Reliability is often specified using an attribute sampling plan for proportions and establishing a target reliability rate (1 – failure rate) and a confidence level (probability of the reliability rate being correct). Because product failures are probabilistic in nature, testing a single unit or running a single test may not surface a failure. To achieve high confidence in the results you must either test a low number of units well beyond target life (impacting schedule), or you must test many units (impacting cost).

The table below shows some example reliability calculations using an attribute sampling plan for proportions. This table shows for a given confidence level, the reliability rate that can be demonstrated, decreases with the number of failures; as well, the higher the reliability rate to be demonstrated, the larger is the sample size required for testing.

Reliability	Confidence	Sample Size	Allowable Failures
95.0%	90%	45	0
91.6%	90%	45	1
46.4%	90%	3	0
19.6%	90%	3	1
84.8%	90%	11	0
74.9%	90%	11	1

ABOUT THE AUTHORS

Theresa Ramirez is a Senior Quality Engineer at Simplexity Product Development working out of their San Diego office. Theresa has earned a BS degree in Bioengineering from UCSD. She is also an ASQ Certified Quality Engineer and Certified Medical Device Auditor. Theresa has over 25 years of experience in product development, including 10+ years as a Quality Lead for medical device products.

Doug Harriman is the Chief Technology Officer of Simplexity Product Development. Simplexity is a Design Engineering Firm as defined by the models in this paper. Doug has over 25 years' experience in the development of mechatronic systems and feedback controls for products ranging from low-volume wind turbines to high-volume consumer 2D and 3D printers.

To LEARN MORE about Simplexity, review [Simplexity's Product Development Process](#) or [contact them about your next design engineering project](#).

<http://www.simplexitypd.com>

© 2021. Simplexity Product Development. All rights reserved.