# Requirement

## 1 Definition

Requirement is the precise statement of **what, how well and under what conditions** something must be done. It is delineated by **capabilities, characteristics and constraints**. There are many kinds of requirements. It can be a contractually binding statement, or a documentation of problem space, or the mean we use to communicate.

## 2 Importance of Requirements

Requirements problems are the single biggest cause of project problems. Wrong requirements mean wrong design. Requirements drive cost, design, schedules, skills, verification plans and operational procedures, literally everything.

## 3 Requirements Analysis

### 3.1 Objectives

Transforms identified top-level needs and objectives into system functional and performance requirements that specifically define:

1. What the system must do
2. How well it must perform
3. What must be accomplished to ensure operational capabilities
4. How interfacing with other systems is met

### 3.2 Requirement Analysis Inputs

1. Customer Needs/Objectives, including: Missions, Environments, Constraints
2. Technology Base
3. Output Requirements from Prior Development Effort
4. Program Decision Requirements/Company constraints
5. Requirements Applied through Specifications and Standards

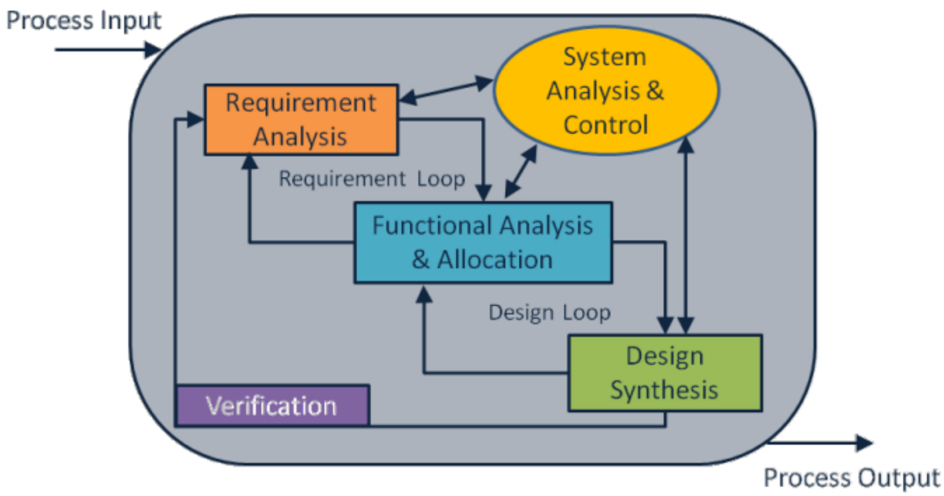


Figure 1: Requirement Analysis.

### 3.3 Requirement Analysis Activities

1. **Eliciting Requirements:** Translating needs and scoping into requirements.
2. **Documenting Requirements:** Including use cases, specifications, lists…
3. **Analyzing Requirements:** Ensure they are clear, complete, valid, consistent

### 3.4 Types of Requirements

1. **Functional:** Requirements which define what an item must do.
2. **Performance:** Requirements which define and quantify how well an item must accomplish a particular function.
3. **Constraints:** Requirements that capture operational, environmental,interface, safety or regulatory constraints.
4. **Verification:** Requirements capture how confidence will be established that the system will perform in its intended environment.

## 4 "Requirements" of Requirements

### 4.1 Overview

1. We must capture **All** the necessary requirements.
2. We must have **no frivolous requirements**.
3. We must ensure we **MEET** requirements.

### 4.2 Good Requirements are SMART

1. **S**pecific: It must address only one aspect of the system design or performance and it must be expressed in terms of the need (what and/or how well), not the solution (how).
2. **M**easurable: Performance is expressed objectively and quantitatively.
3. **A**chievable: It must be technically achievable at costs considered affordable.
4. **R**elevant: It must be appropriate for the level being specified.
5. **T**raceable: Lower level requirements (children) must clearly flow from and support higher level requirements (parents).

## 5 Requirements Management

### 5.1 Change Requirements

Even though normally we don't want to change requirements during the project, it is still possible to have some changes due to:

1. Change of Market
2. Change of Architecture
3. Change of Design Artifact

### 5.2 Create Requirements

Requirements are also possible created during the project, through:

1. **Flowdown:** What applies to the larger context, applies to the smaller
2. **Derivation:** New knowledge/design decision is codified in a requirement
3. **Allocation:** Assign requirements to functions or parts

## 6 Quality Function Deployment (QFD)

### 6.1 Introduction

QFD technique is a systematic approach of translating a societal need or a military threat to program specific goals. It consists of techniques for creating and completing a series of matrices showing the association between specific features of a product and statements representing the customer needs.

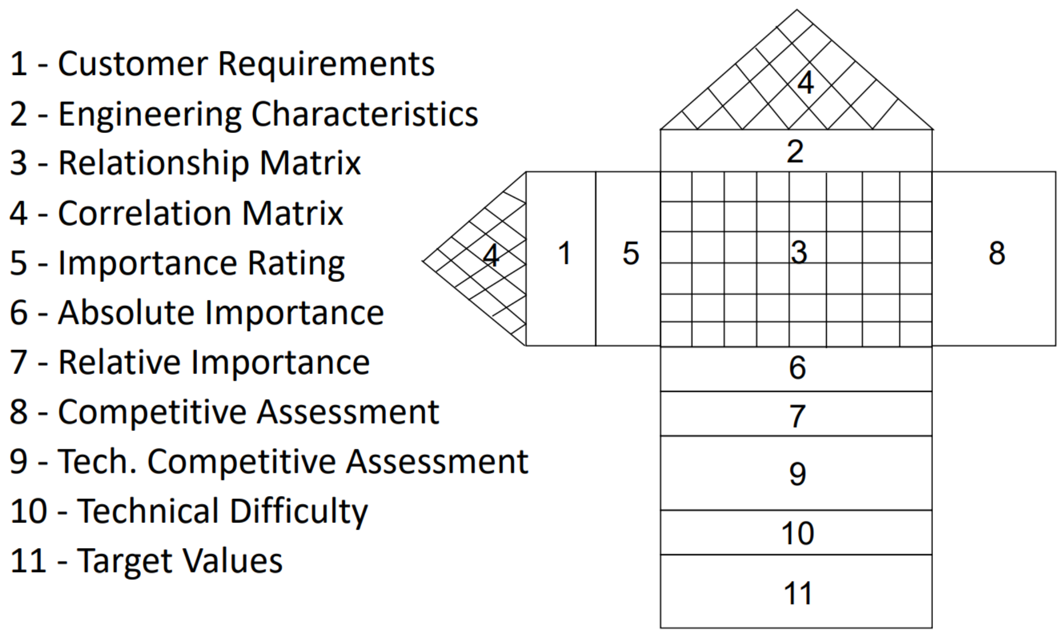


Figure 2: QFD Elements

## 6.2 Key Elements in QFD

### 6.2.1 Customer Requirements

Frequently referred to as the **whats** of the QFD. There are 3 categories:

1. **Spoken Requirements:** Directly indicated by the customer.
2. **Hidden Requirements:** Implicit customers' or stakeholders' needs.
3. **Exciters:** "Softer" customer requirements that can be traded off against other needs.

### 6.2.2 Engineering Characteristics

Frequently referred to as the **hows** of the QFD. They identify how the customer requirements can be satisfied.

### 6.2.3 Relationship Matrix

This part is the body of the QFD. It identifies the correlation between the **hows** and the **whats**, usually labeled as weak, medium or strong or assigned an associated quantitive value.

1. **Empty Column:** An engineering characteristic previously thought significant does not have an impact on any of the customer requirements.
2. **Empty Row:** A customer requirement is not being properly addressed with the current set of engineering characteristics.

### 6.2.4 Correlation Matrix

Identify the relationships that exist among either the customer requirements or engineering characteristics. Usually, it is labeled as positive, strong positive, negative, or strong negative. This is important so that possible trades can be recognized early in the design process when the cost incurred is low and changes in the design are easily made.

### 6.2.5 Importance Rating

This part shows the importance of each customer requirement. Ratings are established by taking the results from the problem definition phase and allocating them along a given scale.

### 6.2.6 Absolute Importance (Calculated)

This part is obtained by multiplying the quantitative value in each of the cells of the Relationship Matrix by the respective importance rating. The resulting values are then summed for each column in the Relationship Matrix to produce the absolute importance.

### 6.2.7 Relative Importance (Calculated)

Absolute importance on a normalized scale from 1 to 100. This part facilitates the quick identification of the most significant engineering characteristics for the design problem.

### 6.2.8 Customer Assessment

This part shows how the top few competitive products rank with respect to the customer requirements.

### 6.2.9 Technical Assessment

This part benchmarks the company performance against the same few competitor products for each of the engineering characteristics. It allows the decision-maker to discern the best places to allocate resources in order to out-perform the competition.

### 6.2.10 Technical Difficulty

This part shows the ease with which each engineering characteristic can be achieved using a numerical scale. The assigned values are based on estimates by the IPT for the probability of achieving the target values.

### 6.2.11 Target Values

This part contains goals set for each engineering characteristic. The design should be evaluated with respect to these targets throughout the design process.

### 6.3 QFD Interpretation

1. A simple method of showing importance is a bar chart of the calculated importance of each "How".
2. Scan the relationship matrix for any blank rows or columns, which will show a gap in meeting the customer requirements.
3. Find strong negative relationships in the correlation matrix, this highlights trade-offs.
4. Highly important and highly difficult to meet technical characteristic targets imply a risk.

## 7 Management and Planning Tools for QFD

1. **Affinity Diagram:** Creative method of loosely grouping issues using a bottom-up approach.
2. **Tree Diagram:** Logical method of decomposing a problem or solution using a top-down approach.
3. **Interrelationship Digraph:** Graphical tool determining importance by drawing lines of cause and effect.
4. **Prioritization Matrix:** Logical method of ranking issues and alternatives.
5. **Matrix Diagram:** Shows the correlation between each idea in one group of items to ideas in other groups.
6. **Process Decision Program Chart:** Creative method to identify barriers and potential alternatives.
7. **Activity Network:** Flowchart tools to compare alternative processes and reduce slack time.