System Engineering and Analysis 2
(Conceptual System Design)

Nan Tu PhD

Spring, 2007

Tsinghua University
Industrial Engineering Department

Agenda

- Basic Conceptual System Design Concept
- Quality Deployment Function

- Reading Material: Heuristics

Not all Systems are Designed

- GorTex (a water-prove material) was **invented**, not designed.
- The North American river systems **evolved**. Mammals **evolved** from fish, not designed.
- Coral reefs **grow**. The Internet **grew**, and its phenomenal growth was neither designed nor expected.
- Systems do not have to be designed by humans. A beaver dam is designed by **beavers**.

System Experience

"Experience is the hardest kind of teacher. It gives you the test first and the lesson afterward"

Why are Mistakes Good?

- Success comes from wisdom
- Wisdom comes from experience
- **Experience comes from mistakes**
- In real life, you **learn** from the mistakes.

Class Discussion

- Question: how many times did Edison try his light bulbs?

Class Discussion

- Question: how many times did Edison try his light bulbs?
- From 1878 to 1880, reportedly

> 3,000.

Engineering Team

25055 -- R&D Systems Engineer

Company: Cardiac Rhythm Management

The Systems Engineer is responsible for designing, defining and implementing system requirements for complex customer systems and or preparing studies and analyzing existing systems. Key Responsibilities: Designs, defines, implements system requirements, system specifications, input output processes and working parameters by using current systems engineering methods and technologies. Documents system requirements by writing documents, reports, memos, change requests. Methods used are determined by approved procedures and standards. Tracks system engineering effort by creating and maintaining records in the approved tracking management tool. This person will be responsible for analyzing returned products, answering questions from the field, and implementing improvements to released products. BS and/or MS degree required plus 3+ years of relevant systems work experience.

- How to pick a **winning team**?
- Job Description/ resume
  - Name
  - School / Degree
  - Years Experience
  - Specialties
  - Other trainings

Class Discussion:

- A CD Player **Design and Manufacturing** Team

- A Mechanical Engineer
- A Electrical Engineer
- An Industrial Designer
- A Manufacturing Engineer
- A Program Manager
- Anybody else?

The Three Laws of Systems Engineering

- Everything Interacts with Everything Else
  - **Decomposition**
- Everything Goes Somewhere
  - **Interfaces**
- There is No Such Thing as a Free Lunch
  - **Trade-off studies / Decision analysis**

Conceptual Design

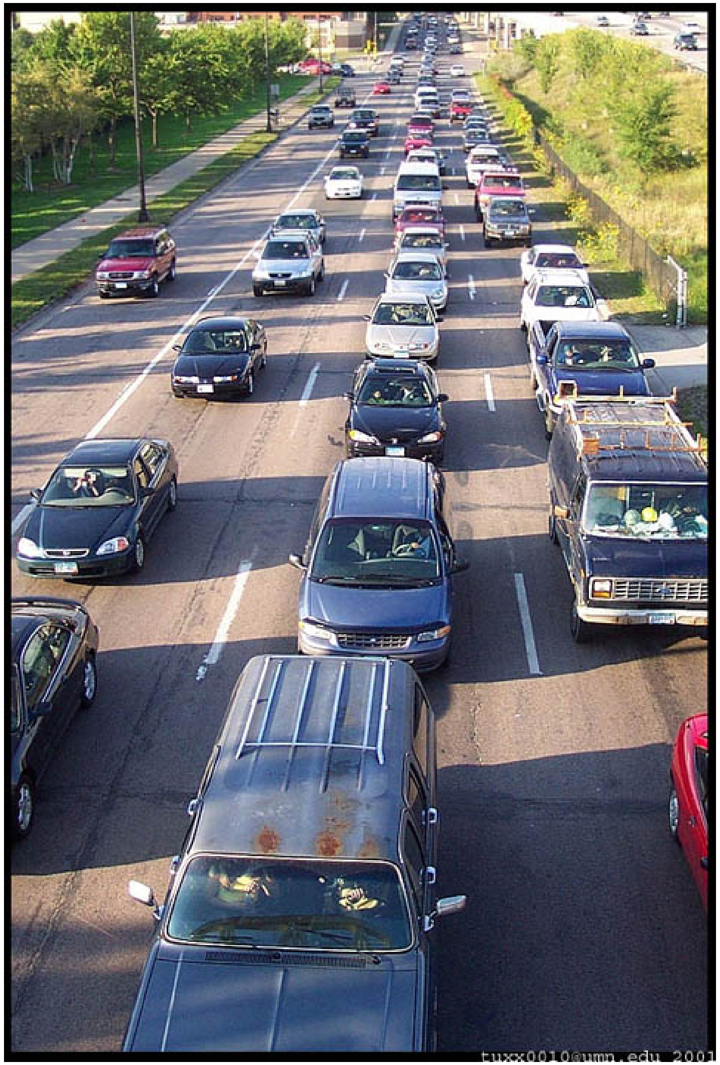
- The **beginning** is the most important part of the work.
  - Plato, 4th Century, B.C.

Identification of Need

- Conducted to gain an understanding of the **System Requirements**
- **Defining** the problem is the most difficult task in the system engineering process

Class Discussion - Need

- Traffic situation in Beijing. What is the problem?
  - Problem: Daily traffic jam during rush hours 7:30 – 10:00 am and 4:30 – 7:00 pm
  - **Need:** What are the customer's needs for the road?
    - reduce the time it takes to go from point A to point B?
    - ……..



Traffic situation – Possible Solutions

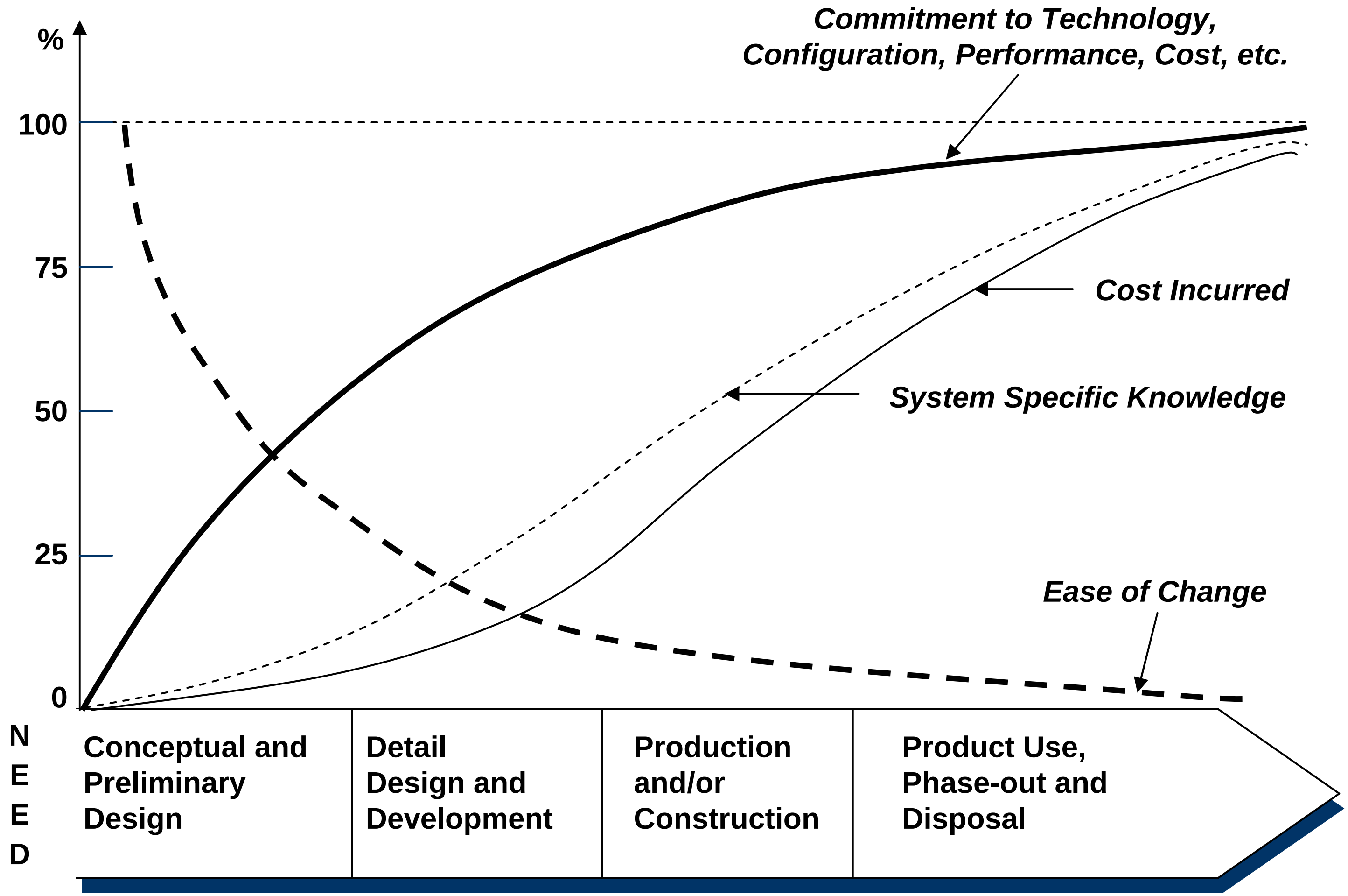
- Traffic Jam is one of the **oldest** problem since the invention of cars.
- Possible Solutions:
  - More roads
  - Public transportation, ie: buses, subways, rapid transit
  - Man powered systems, ie. bikes
  - Government restrictions, cars with even numbers can enter the city at a given day
  - Economic incentives, pay to enter city
  - Automated vehicles,
  - …….

Refer to Scientific America 1997 Oct issue

Feasibility Analysis

- Identify possible **system – level** design approaches
- Evaluate the most **likely** approaches
- **Narrow** the number to be a few
- Recommend a **preferred** course of action

Implementing Systems Engineering



System Requirement Analysis (SRA)

- **Operational** Requirements
- **Maintenance and Support** Requirements
- Technical **Performance** Measures (TPMs)
- **Functional** Analysis and Allocation (System Level)
- Synthesis, Analysis and Evaluation

SRA True of False

Is done only at the beginning of the project

Can be applied to buying a home

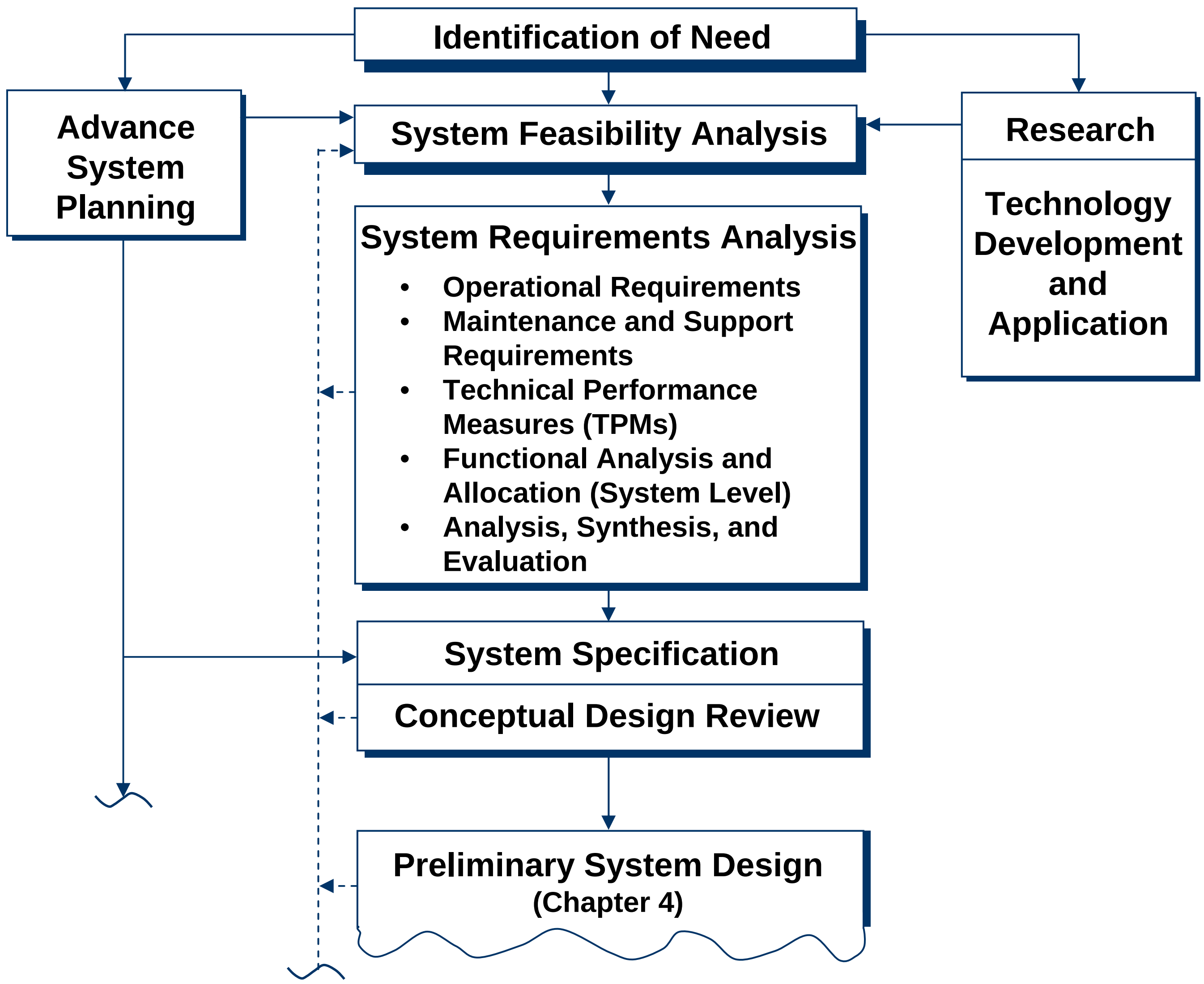
Must be applied to all projects

Must be recorded on all projects by the Systems Engineer down to the component level

SRA True of False

- ✓ Is done only at the beginning of the project
- ✓ Can be applied to buying a home
- ✓ Must be applied to all projects
- ✓ Must be recorded on all projects by the Systems Engineer down to the component level

System Requirements Definition



System Specification Class Discussion

- A CD Player must be potable
- Users must be able to tell when the CD Player is on/off
- ……..
  - Please have two students, each write the requirements on the blackboard. Please also rand the needs on the scale of 1-5. 5 very important, 1 not very important.

System Specification a CD Player

- A CD Player must be potable
- The CD Player must be resistant to active/rigorous motion.
- Users must be able to tell when the CD Player is on/off.
- The CD Player must allow for insertion and removal of the CD.
- The CD Player must work with a headphone sets.
- The CD Player must work with home stereo systems.
- The CD player must have a good interface
- ……

System Specification Class Discussion: a Soda Vending Machine

- The Soda Vending Machine must allow a customer buy a soda.
- ....

System Specification Class Discussion: a Soda Vending Machine

- The Soda Vending Machine must allow a customer buy a soda.
- The Soda Vending Machine must allow a supplier to restock sodas and change.
- The Soda Vending Machine must allow a supplier to collect the accumulated money from the soda vending machine.
- ….

Operational Requirements

- **Mission profile** or scenario
- **Performance** and related parameters
- **Utilization** requirement
- Effectiveness requirements
- Operational life cycle
- Environment

Operational Requirements

- **What** functions will the system perform?
- **When** will the system be required to perform its intended function and for how long?
- **Where** will the system be used?
- **How** will the system accomplish its objective?

Maintenance and Support Requirements

- Levels of maintenance
- Repair policies
- Organizational responsibilities
- Logistic support elements
- Effectiveness requirements
- **Environments**

Quality Function Deployment (QFD)

- A **team approach** to ensure that the voice of the customer is reflected in the ultimate design
- Quality Function Deployment (QFD) translates **decision criteria** or Critical-To-Quality issues into a prioritized set of targets, choices, or improvement opportunities - helping you to produce better products, processes, services, or strategies.

Quality Function Deployment (QFD)

- May be used as a tool to support SRA and Mission Analysis.
- Provides a **means** by which to translate the Customer's requirements into the appropriate technical requirements needed to provide the Customers wants (in QFD these are known as "WHATS")

The QFD Matrix

- Helps put the requirements at the top level and attributes of the implementation on a single piece of paper
- Correlation analyses can be done
- Attribute balancing and prioritization can be done
- Output becomes part of the Development Specification

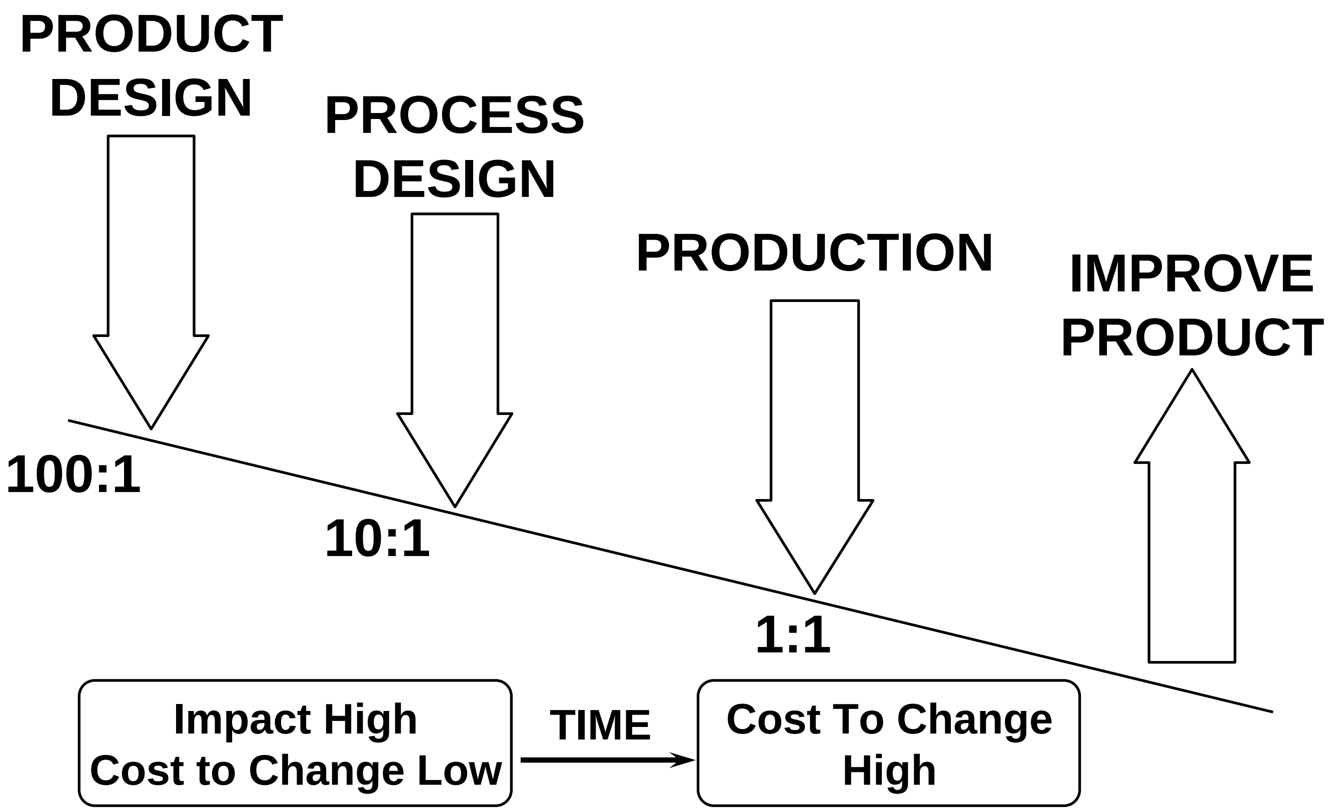
QFD Goals

- Establish **who** the customers are and then determine **what** the customers want (WHATS)
- Determine how to satisfy the Customer's WHATS. (In QFD these are referred to as "**HOWS**")

QFD Steps (detailed steps)

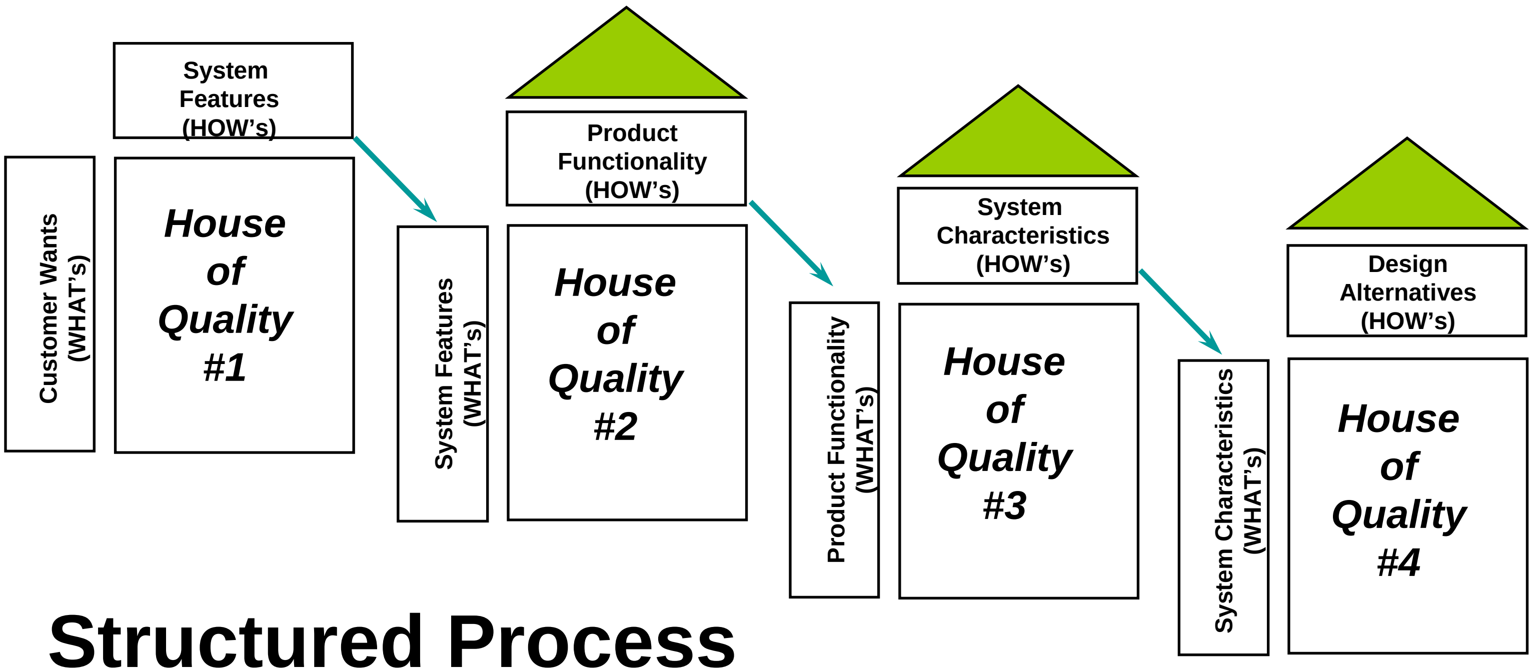
- Derive top-level product requirements or technical characteristics from customer needs (Product Planning Matrix).
- Develop product concepts to satisfy these requirements.
- Evaluate product concepts to select most optimum (Concept Selection Matrix).
- Partition system concept into subsystems or assemblies and flow-down technical characteristics to these subsystems or assemblies.
- Derive lower-level product requirements (assembly or part characteristics) and specifications from subsystem/assembly requirements (Assembly/Part Deployment Matrix).
- For critical assemblies or parts, flow-down lower-level product requirements (assembly or part characteristics) to process planning.
- Determine manufacturing process steps to meet these assembly or part characteristics.
- Based in these process steps, determine set-up requirements, process controls and quality controls to assure achievement of these critical assembly or part characteristics

Design – Quality Trade-off (QFD)



The Quality Lever

How Does QFD Work?

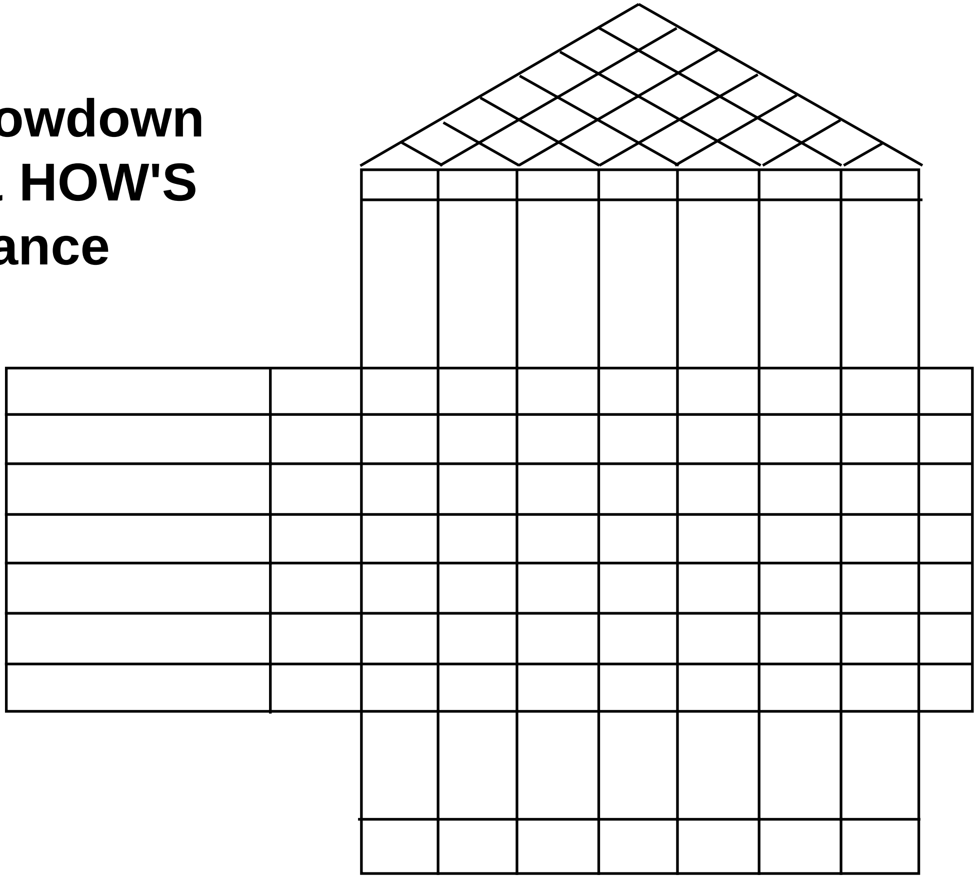


- **Structured Process**
  - **Identifies CTQs At One Level**
  - **Relates Those CTQs To The Next Level**

The Basic Building Block Is The House Of Quality

The Basic House of Quality

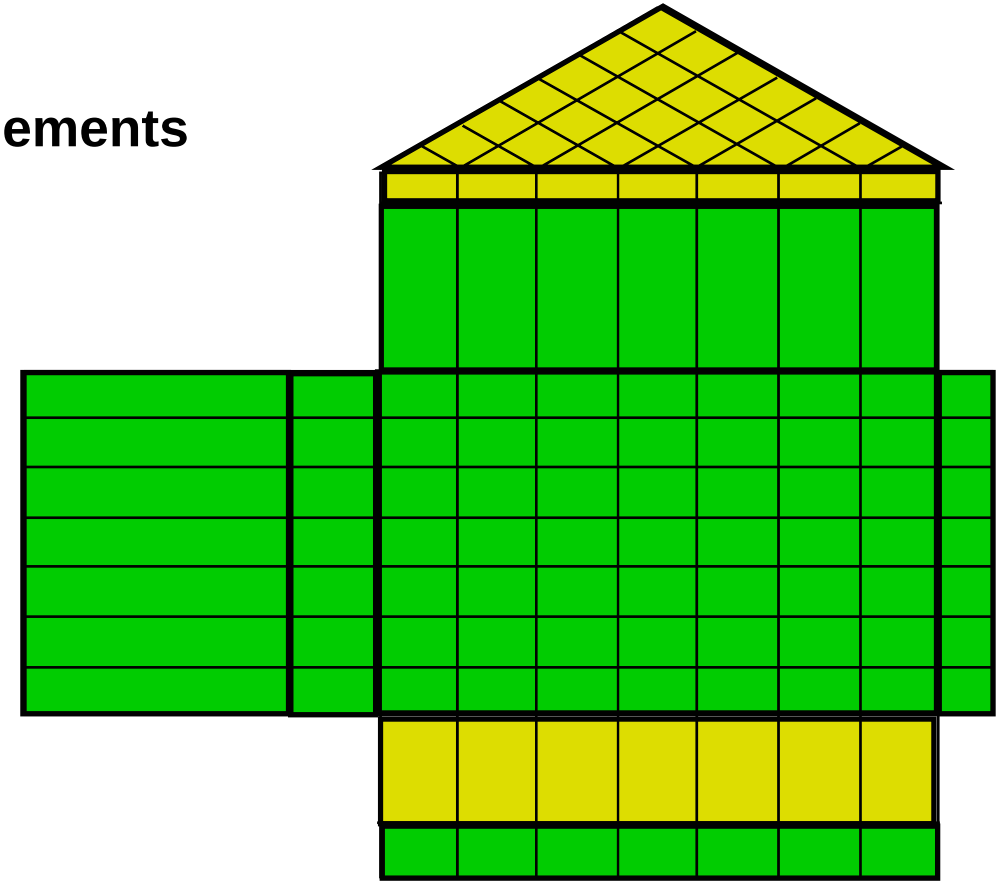
- Establishes the Flowdown
- Relates WHAT'S & HOW'S
- Ranks The Importance



The Basis of QFD is the House

The Basic House of Quality

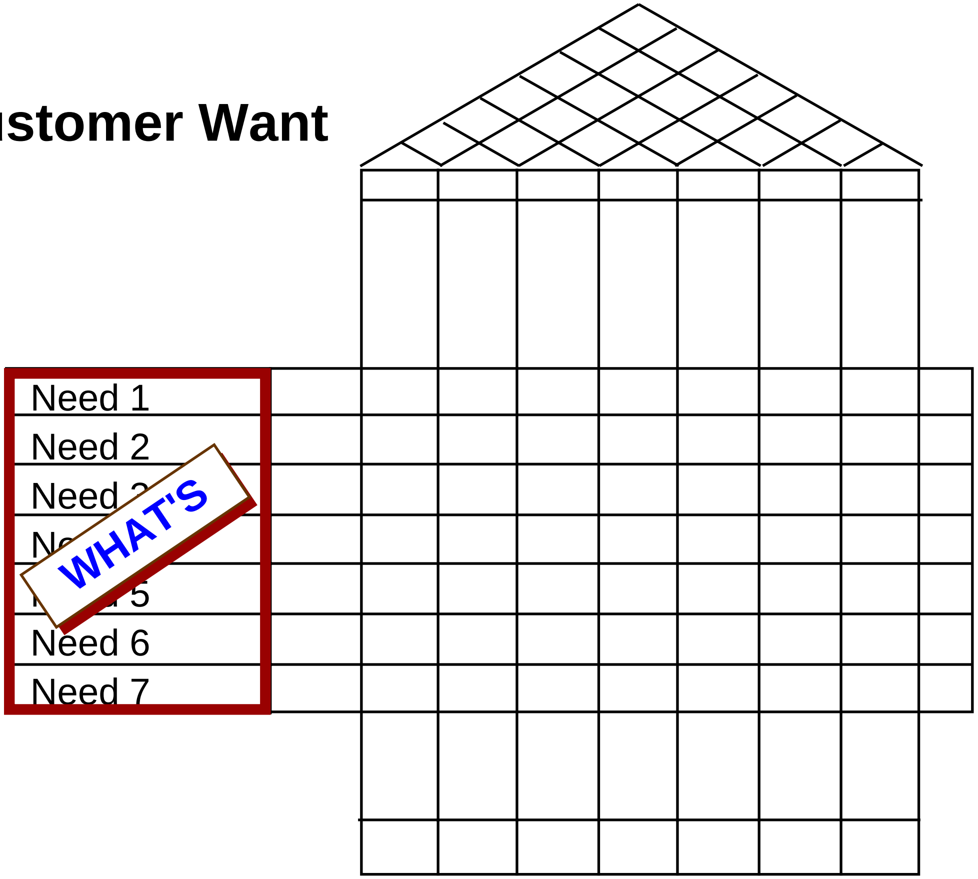
Key Elements
Informational Elements



Two Element Types In Each House

Key Elements - "WHAT'S"

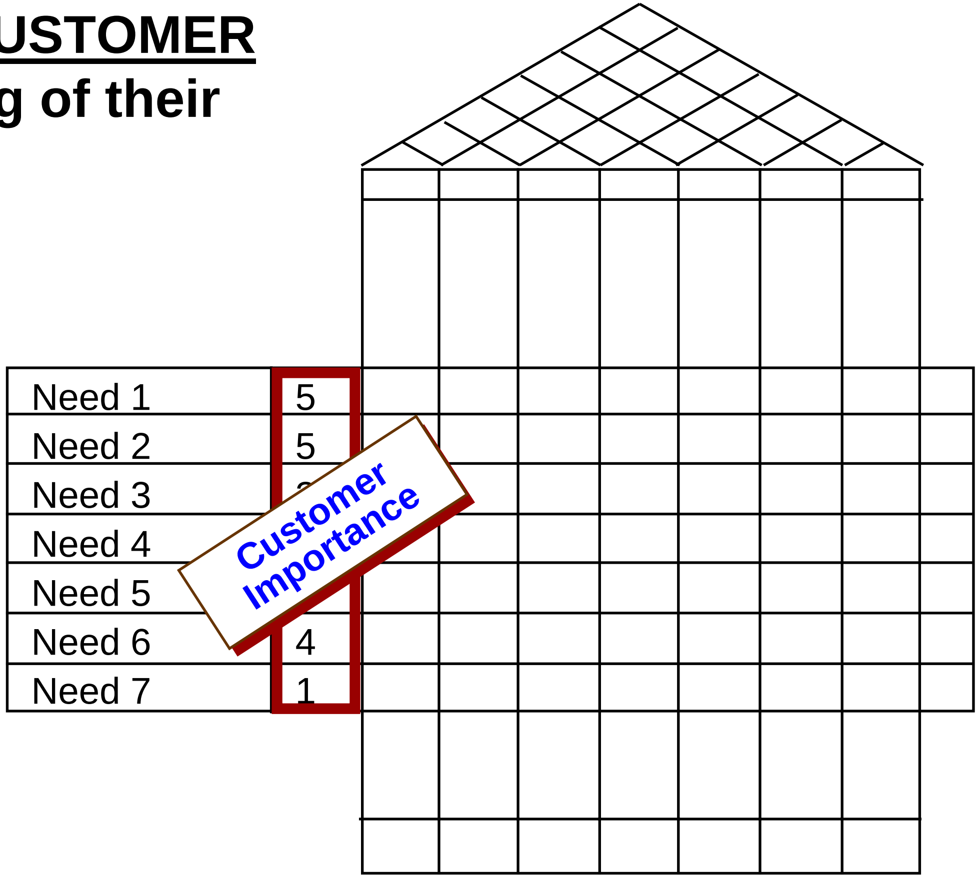
- What Does The Customer Want
- Customer Needs



Voice of the Customer

Key Elements – Customer Importance

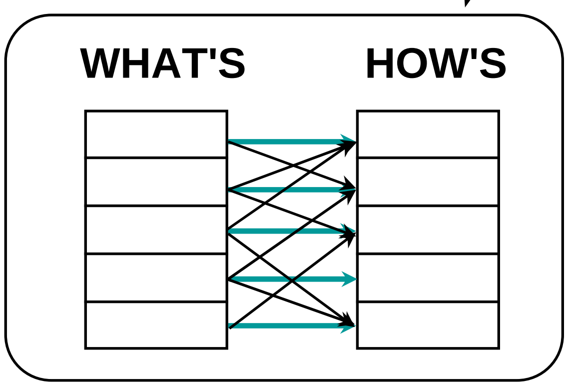
- **How Important Are The What's <u>TO THE CUSTOMER</u>**
- **Customer Ranking of their Needs**

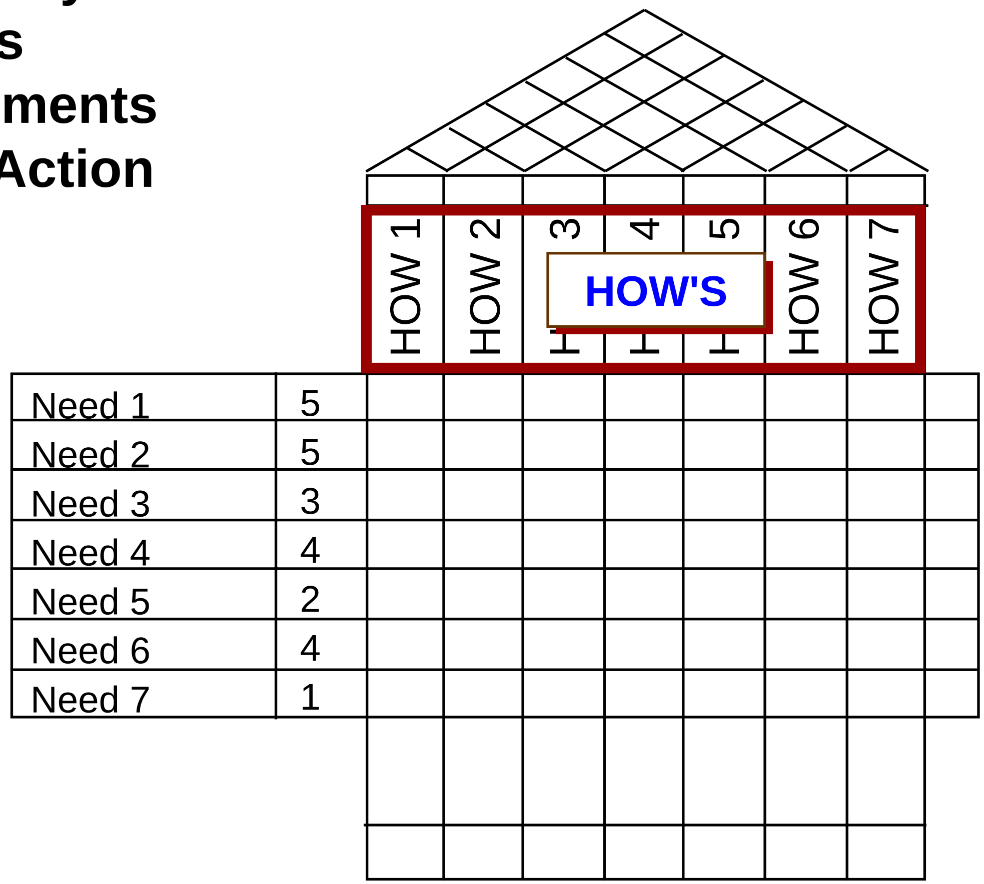


Voice of the Customer

Key Elements - “HOW’S”

- **How Do You Satisfy the Customer What’s**
- **Product Requirements**
- **Translation For Action**
- **Xs**

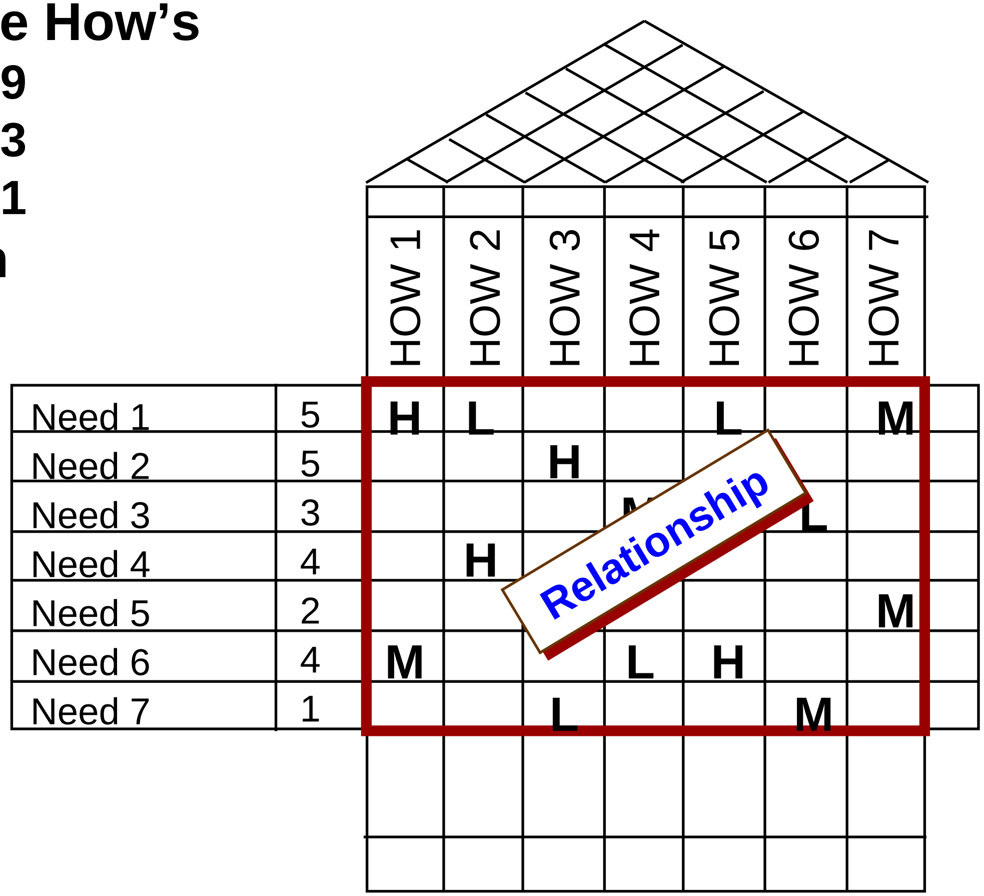




Satisfy the Customer Needs

Key Elements - Relationship

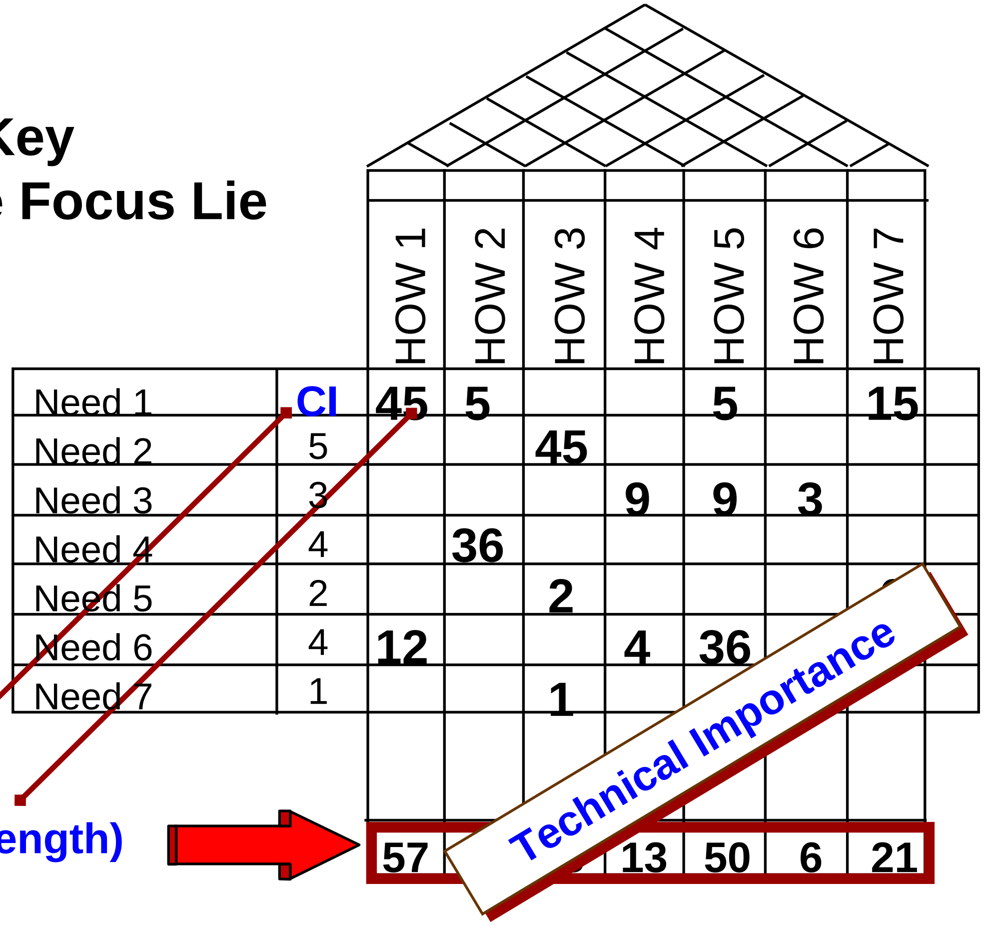
- **Strength of the Interrelation Between the What's and the How's**
  - **H Strong 9**
  - **M Medium 3**
  - **L Weak 1**
- **Transfer Function**
- **Y = f(X)**



Untangling The Web

Key Elements – Tech. Importance

- **Which How's are Key**
- **Where Should The Focus Lie**

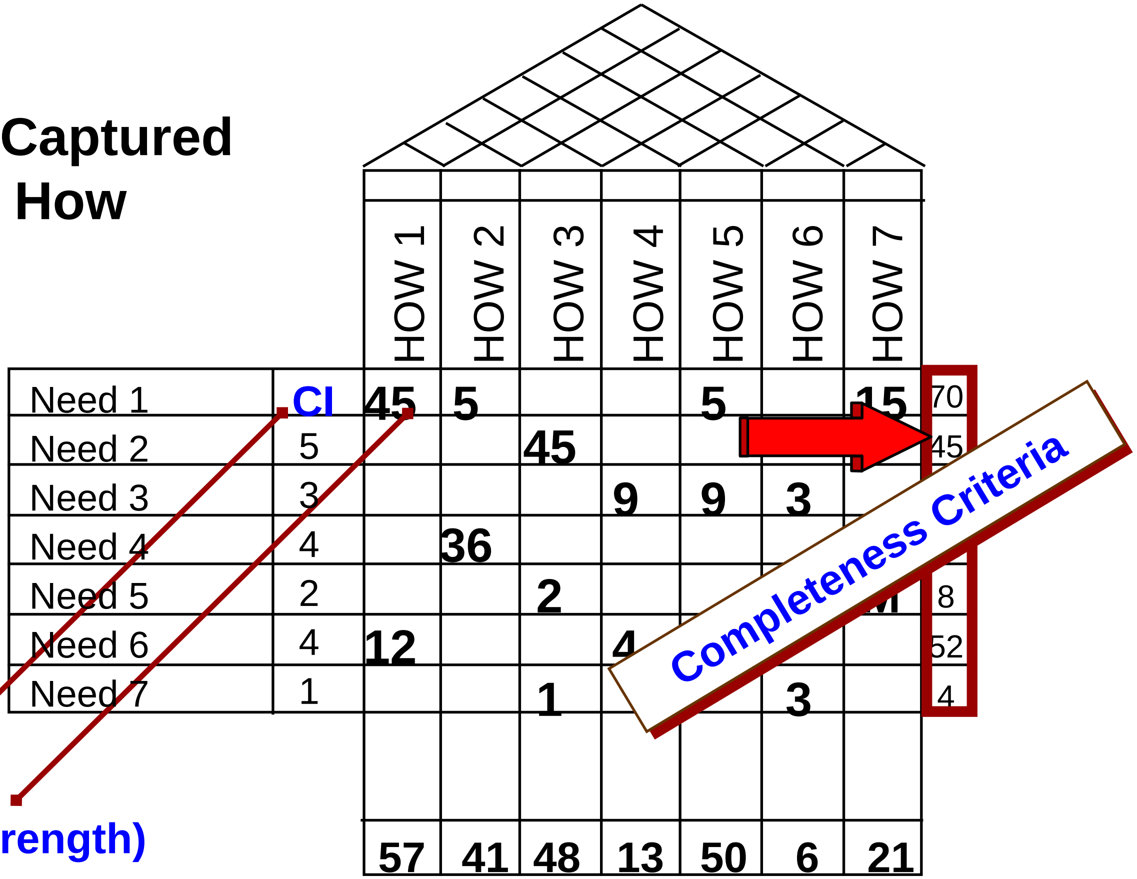


- $\mathbf{TI} = \sum_{column} (\mathbf{CI} * \mathbf{Strength})$

Ranking The HOW'S

Key Elements - Completeness

- **Are All The How's Captured**
- **Is A What Really A How**

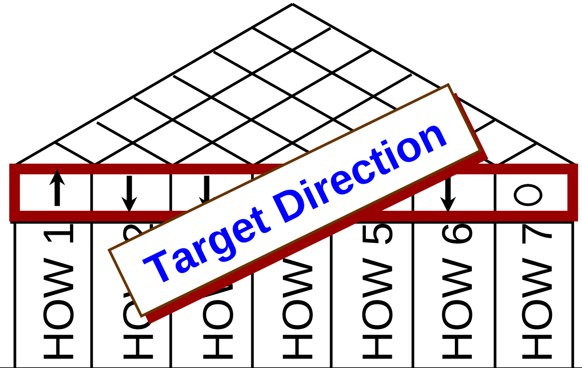


- $CC = \sum_{row} (CI * Strength)$

Have We Captured the HOW'S

Information – Target Direction

- **Information On The HOW'S**
  - ↑ **More Is Better**
  - ↓ **Less Is Better**
  - ○ **Specific Amount**



		HOW 1	HOW 2	HOW 3	HOW 4	HOW 5	HOW 6	HOW 7	
Need 1	5	H	L			L		M	65
Need 2	5			H					45
Need 3	3				M	M	L		21
Need 4	4		H						36
Need 5	2			L				M	8
Need 6	4	M			L	H			52
Need 7	1			L			M		4
		57	**41**	**48**	**13**	**50**	**6**	**21**	

The Best Direction

Information – How Much

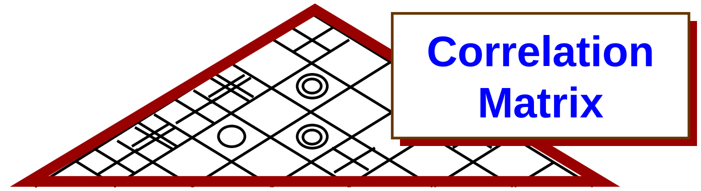
- Target Values for the How's
- Note the Units

		HOW 1	HOW 2	HOW 3	HOW 4	HOW 5	HOW 6	HOW 7	
		↑	↓	↓	↑	↓	↓	O	
Need 1	5	**H**	**L**			**L**		**M**	65
Need 2	5			**H**					45
Need 3	3				**M**	**M**	**L**		21
Need 4	4		**H**						36
Need 5	2			**L**				**M**	8
Need 6	4	**M**			**L**	**H**			52
Need 7	1			**L**			**M**		4
How Much		3 lbs	12 in.	3 mils	40 psi	3	8 atm		
		57	**41**	**48**	**13**	**50**	**6**	**21**	

Consistent Comparison

Information – Correlation Matrix

- **Impact Of The How's On Each Other**
  - ◎ **Strong Positive**
  - ○ **Positive**
  - ╳ **Negative**
  - ⩙ **Strong Negative**



		↑	↓	↓	↑	↓	↓	○	
		HOW 1	HOW 2	HOW 3	HOW 4	HOW 5	HOW 6	HOW 7	
Need 1	5	**H**	**L**			**L**		**M**	65
Need 2	5			**H**					45
Need 3	3				**M**	**M**	**L**		21
Need 4	4		**H**						36
Need 5	2			**L**				**M**	8
Need 6	4	**M**			**L**	**H**			52
Need 7	1			**L**			**M**		4
		3 lbs	12 in.	3 mils	40 psi	3	8 atm	1 mm	
		57	**41**	**48**	**13**	**50**	**6**	**21**	

Conflict Resolution

QFD Flow Down

Levels Of Granularity

Manufacturing Environment

- Customer Wants
- ↓ Technical Requirements
- ↓ Part Characteristics
- ↓ Manufacturing Process
- ↓ Production Requirements

Software Environment

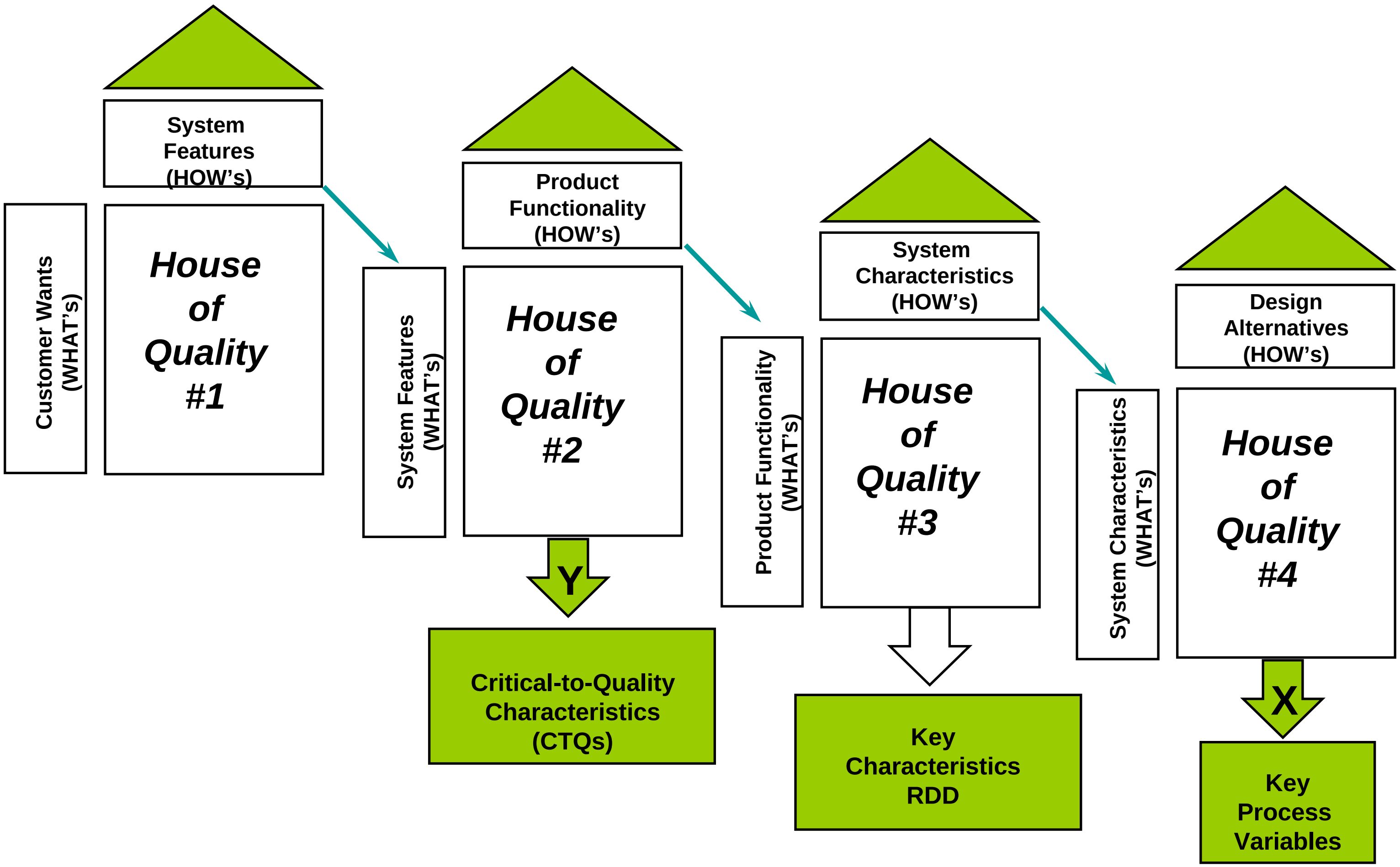
- Customer Wants
- ↓ Product Functionality
- ↓ System Characteristics
- ↓ Design Alternatives

Service Environment

- Customer Wants
- ↓ Service Requirements
- ↓ Service Processes
- ↓ Process Controls

Flowdown Relates The Houses To Each Other

Multilevel QFD



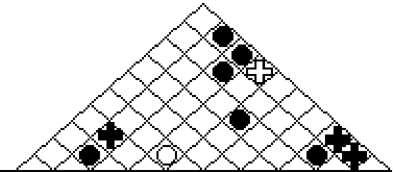
NOTE: The How's at One Level Become the What's at the Next Level

QFD examples

Source: http://www.isixsigma.com/

Auxiliary Power Unit Product Planning Matrix

Interactions:
- Strong Negative (filled cross)
- Moderate Negative (open cross)
- Strong Positive (filled circle)
- Moderate Negative (open circle)



	Customer Needs	Priority	Bleed air ducting location	Maximum APU weight	Low turbine wheel weight	High equivalent shaft horsepower	Controlled turbine inlet temp.	Bleed air	Electrical power output	Turbine assy tri-hub containment	Strong containment ring	Lightweight containment ring	Competitive Evaluation (1-Low, 5-High): 1	2	3	4	5
	Goal		O	↓	↓	↑	O	↑	↑	↑	↑	↓					
	Area		Physical			Operation				Contain							
Interface	Fit with customer envelop/interface	3	5								3				W	T	
Interface	Support oil-cooled generator	5		3											T	W	
Interface	Low weight	4	3	5	3					3		5		T		W	
Oper.	Provide bleed air	4	3			5	5	5						T	W		
Oper.	Provide electrical power	3				5			5					W	T		
Oper.	Operate safely	5			3		3			5	5				WT		
	Reliable	5					5			3					WT		
Technical Evaluation		5															
Technical Evaluation		4			W		T			T		W					
Technical Evaluation		3	W	WT	T	W	W	WT	WT	W	WT						
Technical Evaluation		2				T						T					
Technical Evaluation		1	T														
Target Value / Specification Value			Interface point A	160 lbs.	6 lbs.	350 hp	1850 degrees F	75 lbs/min.	75 KVA	2.5 lbs at power	3 lbs. at power	< 6 lbs.					
Technical Difficulty (1-Low, 5-High)			1	4	3	5	3	3	3	4	2	4					
Importance Rating			39	35	27	35	60	20	15	52	34	20					

W - We
T - They

Summary of today’s class

- Conceptual Design
- SRA (System Requirement Analysis)
- QFD (Quality Function Deployment)

Home Work

- Reading: case studies: CD player, vending machine
- Reading the qualitative vs. quantitative and heuristics

- Not required to submit summary

Additional Reading:

- Qualitative vs. Quantitative Approach
- Mathematical models vs. heuristics

A Designer's Tools

- Engineers Approach their design problems using **analysis and optimization**, powerful and precise tools derived from the scientific method and calculus
- Architects Approach their **qualitative problems** using guideline, abstractions, and pragmatics generated by lessons learned from experience, i.e. heuristics

Qualitative vs. Quantitative

- Quantitative approach
  - Scientific, precise
  - Statistical significance
- Qualitative approach
  - Good enough approach
  - Frequently used in usability studies

Heuristics

- Abstractions of experience, trusted, non-analytic guidelines for treating inherently unbounded, **ill-structured** problems
- Used as aids to decision making, value judgments, and assessments
- Very similar to the ancient wisdom (think of the Chinese proverbs for example)

Heuristic Tool List (1) Multitask Heuristics

- Performance, cost, and schedule depend on the others
- With few exceptions, schedule delays will be accepted grudgingly; cost overruns will not

Heuristic Tool List (2)

- One person's architecture is another person's detail
- In general, each system level provides a context for the level(s) below
- **Social systems**, it's the perceptions, not the facts, that count
- In introducing technological and social change, how you do it is often more important than what you do

Heuristic Tool List (3) Scoping and Planning

- Success is defined by the beholder
- No complex system can be optimum to all parties concerned, **nor all functions optimized**
- The most dangerous assumptions are the **unstated** ones

Heuristic As Tools (4) Modeling

- If you can’t analyze it, don’t build it
- A **model is not reality**
- Constants aren’t and variables don’t
- The eye is a fine architect
- A good solution somehow looks nice

Heuristic As Tools (5) Trades, Options and Choices

- In any resource-limited situation, the **true value** of a given service or product is determined by what one is willing to give up to obtain it
  - How much does an **antique** really cost?
  - Why are they sold in auctions rather than stores?
- If trade results are inconclusive, then the wrong selection criteria were used

Heuristic As Tools – (6) Aggregating

- Group elements that are strongly related to each other, separate elements that are unrelated
- Choose a configuration with **minimal communications** between the subsystems
- System structure should resemble functional structure

Heuristic As Tools (7) Decomposition

- Do not slice through regions where high rates of information exchange are required
- The greatest leverage in architecting is at the interfaces
- Organize **personnel** tasks to minimize the time individuals spend interfacing

Heuristic As Tools (8) - Integrating

- Relationships among the **elements** added value to the **systems**
- Just as a piece and its template must match, so must a system and the resources which make, test, and operate it
- Contain excess energy as close to the source as possible

Heuristic as Tools (9) System Integrity, Quality, and Vision

- As time to delivery decreases, the threat to functionality increases
- Within the same class of products and processes, the failure rate of a product is linearly proportional to its cost
- Mistakes are understandable, failure to report them is inexcusable

Heuristic as Tools (10) Performance Cost, Schedule and Risk

- If you think your design is perfect, it's only because you **haven't shown** it to someone else
- "Proven" and "State of the Art" are mutually exclusive qualities
- The first quick look analyses are often wrong

Heuristic as Tools (11) Evolving, Modifying and Adapting

- The team that created and built a presently successful product is often the best one for its **evolution** - but seldom for creating its **replacement**
- If you don't understand the existing system, you can't be sure you're re-architecting a better one