Concept Generation and Evaluation

MCHE 201 – Spring 2019

Dr. Joshua Vaughan

Rougeou 225

joshua.vaughan@louisiana.edu

@Doc_Vaughan

Phases of Design

1. Problem Understanding
2. Specification Development
3. Conceptual Design
4. Detail Design
5. Production Specification
6. Manufacture
7. Disposal

(Phases 1–3:) What we mainly talk about. The most nonlinear part.

(Phases 4–5:) Machine Design

The entire process is iterative.

Phases of Design

1. Problem Understanding
2. Specification Development
3. Conceptual Design
4. Detail Design
5. Production Specification
6. Manufacture
7. Disposal

(Phases 1–3:) What we mainly talk about. The most nonlinear part.

(Phases 4–5:) Machine Design

The entire process is iterative.

Phases of Design – Tools Used

1. Problem Understanding
   a. Problem Understanding Form
   b. House of Quality

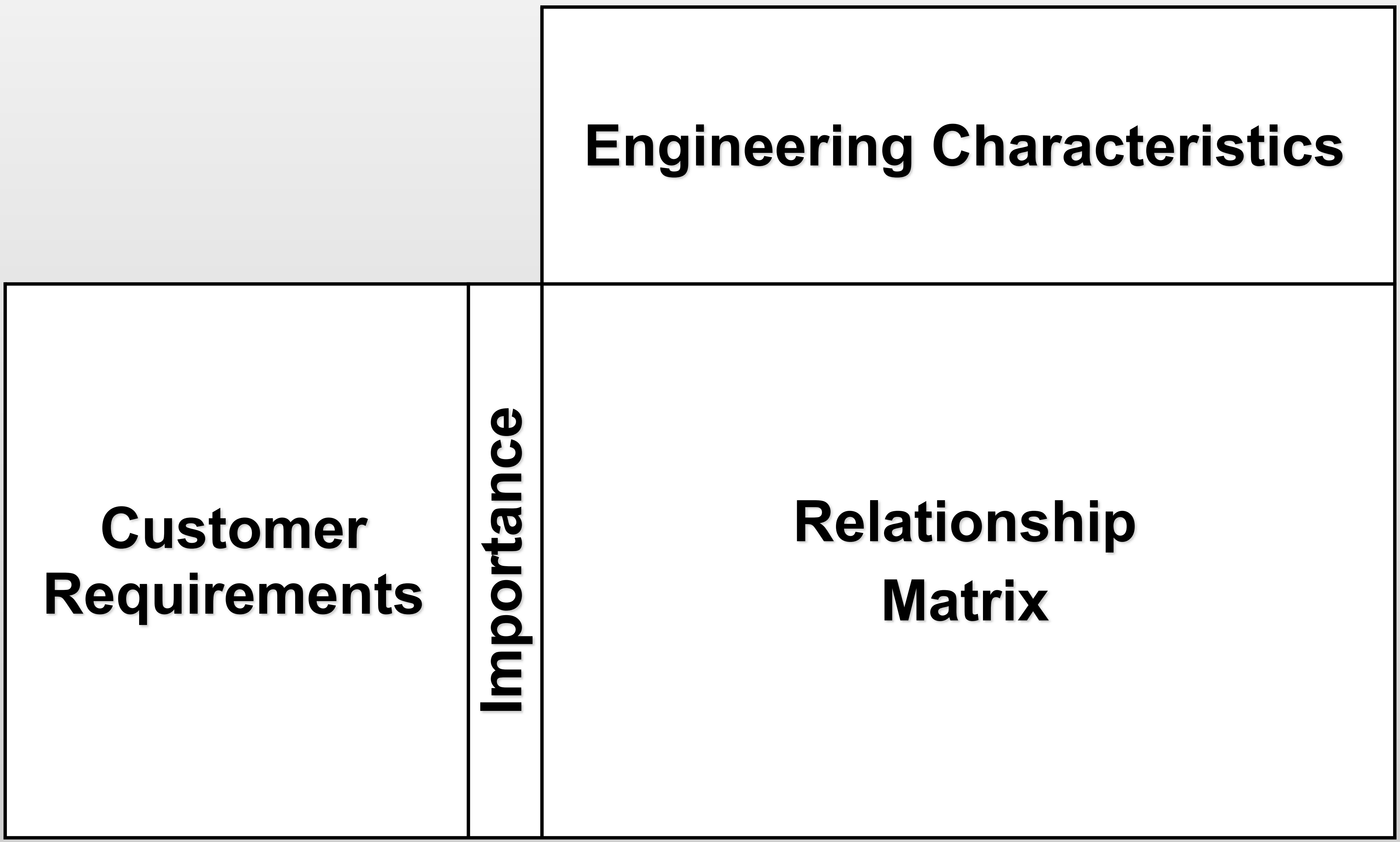
2. Specification Development
   a. Specification Sheet
   b. Function Tree

3. Conceptual Design
   a. Morphological Chart
   b. Concept Evaluation Matrices

The entire process is iterative, including the tools!

The Problem Understanding Form

The “middle” of the House of Quality

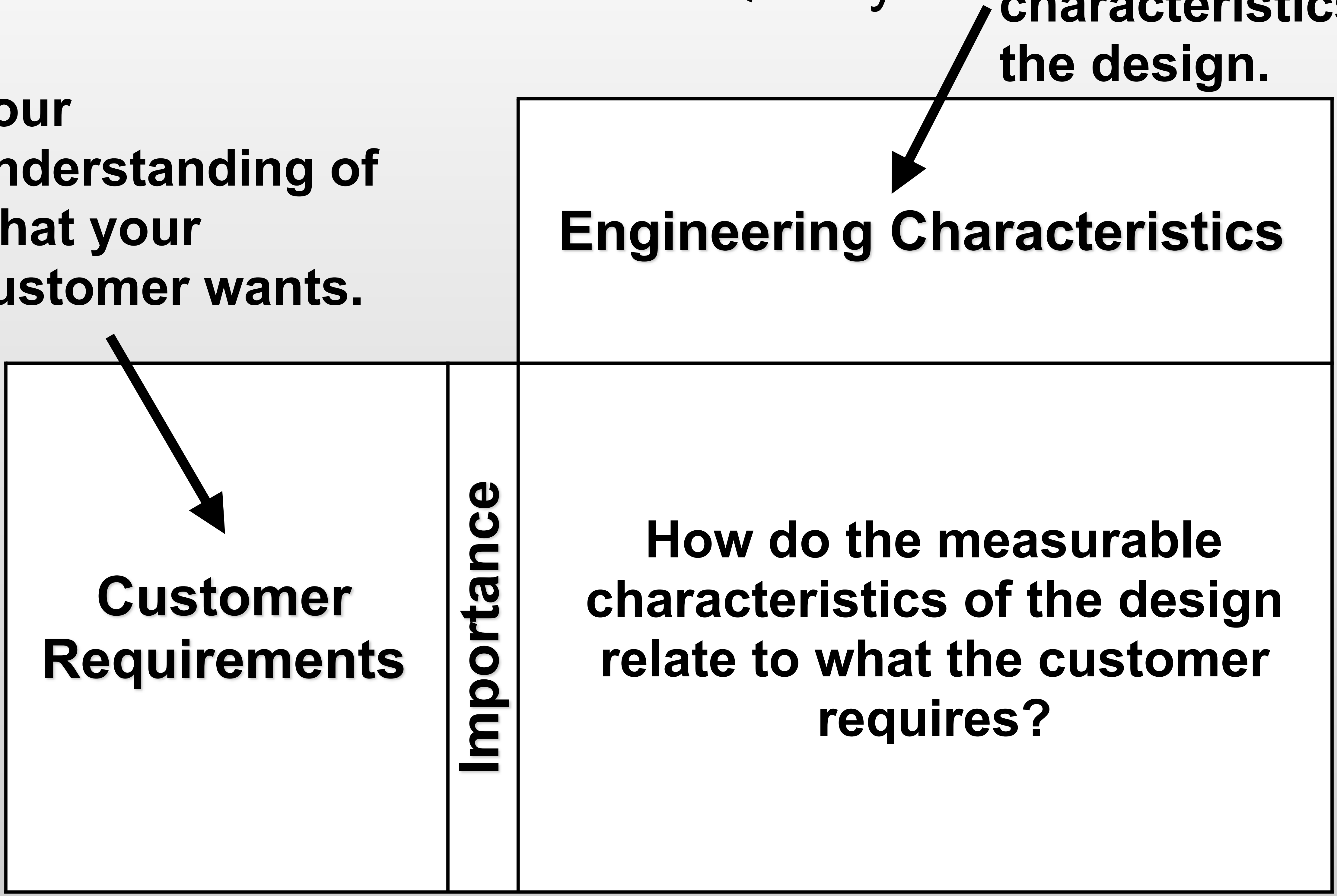


The Problem Understanding Form

The "middle" of the House of Quality

Measurable characteristics of the design.

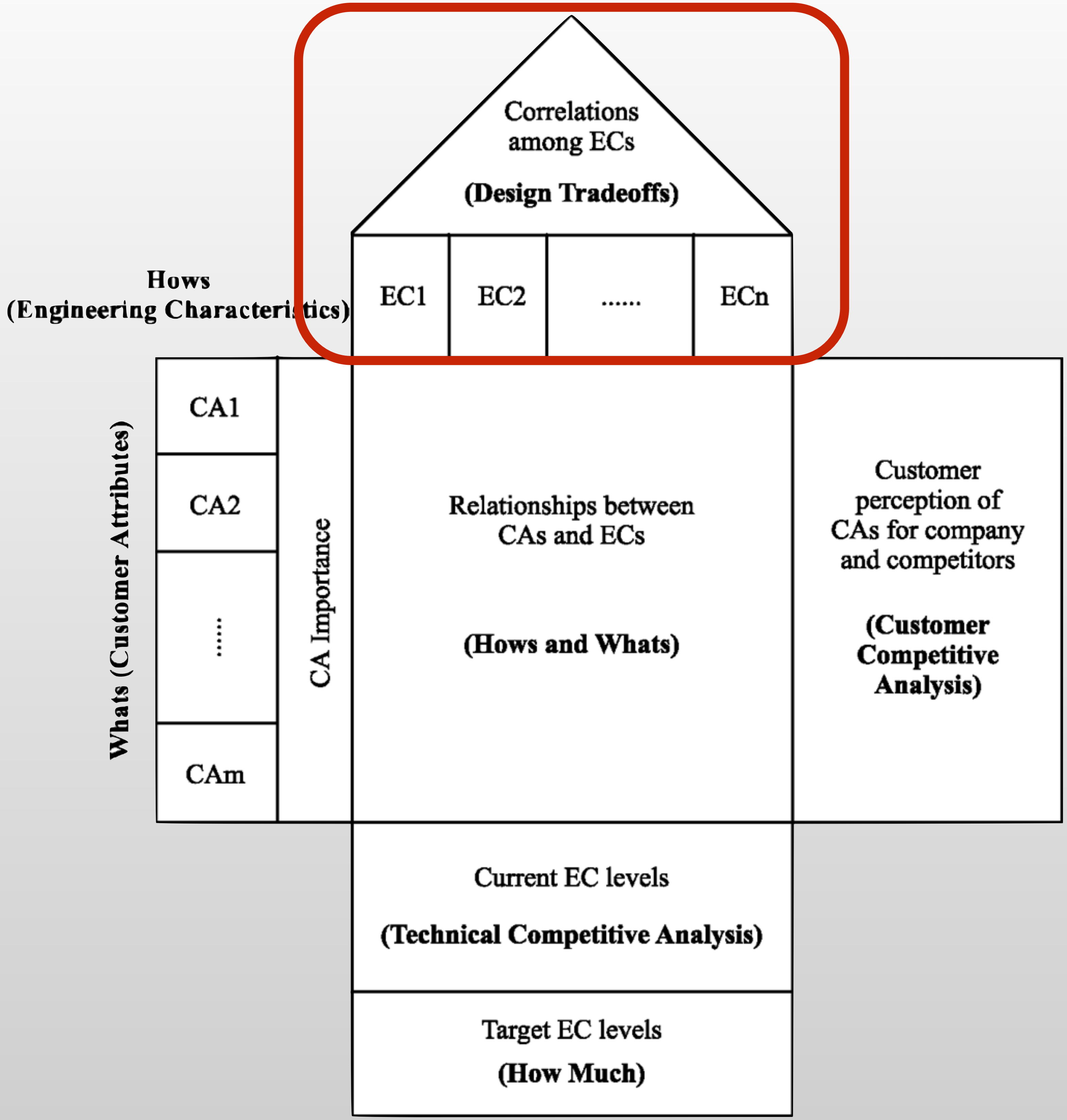
Your understanding of what your customer wants.



House of Quality Tips/Hints

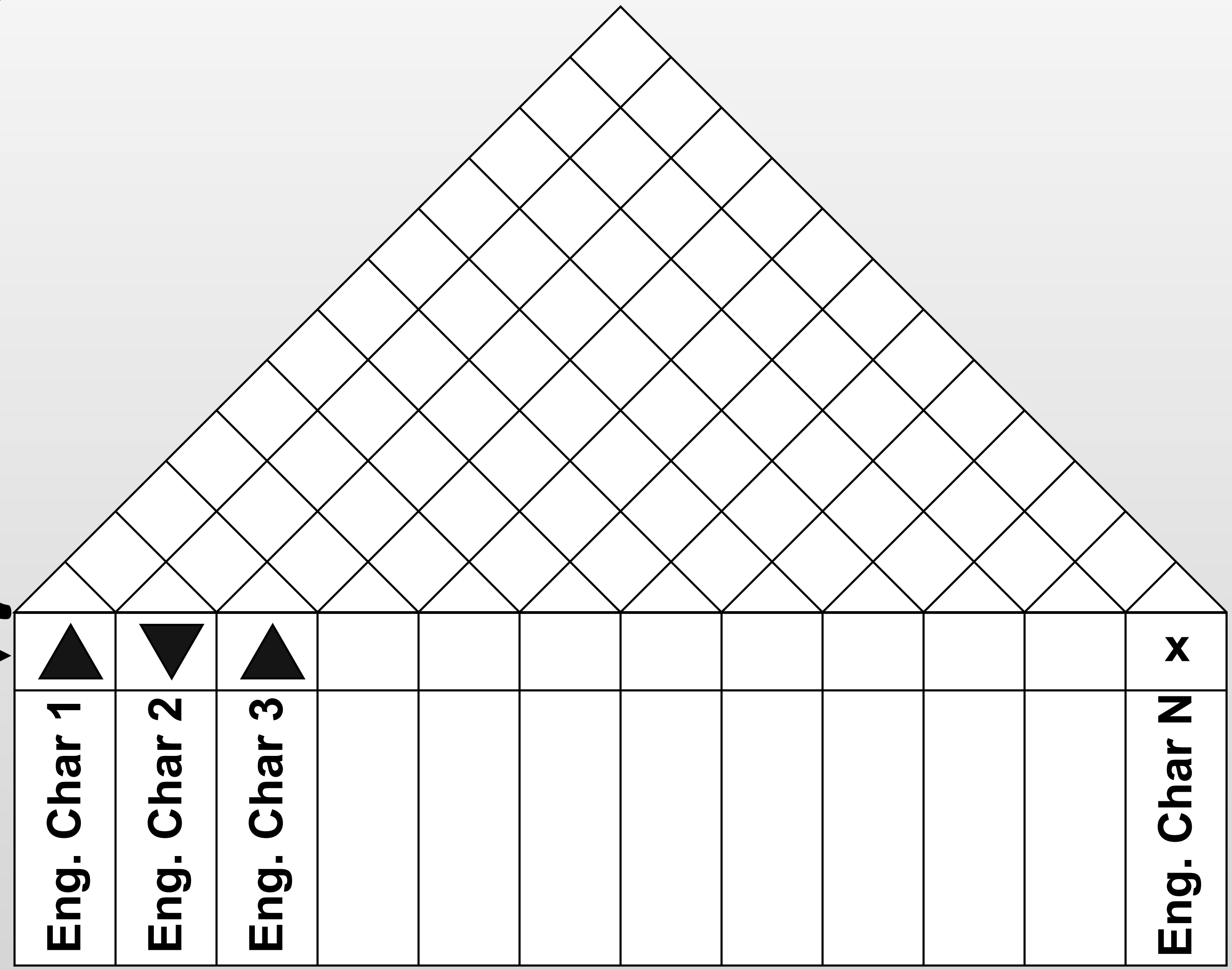
- *Explicitly* identify your customers
- List every thing any of your customers care about
- Take each point of that list and expand
  - What does this *really* mean?
  - What does the customer *really* want?
    - *e.g.* Reliable?… Uptime of 99.99%? Can survive misuse?
    - *e.g.* Easy to use?… Low number of steps? Easily understandable process? Low physical effort needed?
- For each customer requirement, determine what you could measure to determine if you are satisfying it or not.
- Revisit regularly as your understanding of the problem improves!!!

The HoQ Correlation Matrix



The HoQ Correlation Matrix

How do improvements toward the goal affect others?



Goal:

Maximize – ▲

Minimize – ▼

Meet Target – x

The Spec. Sheet

		For: **PRODUCT NAME**	**Issued:** mm/dd/yy Page x of N	
Changes	**D/W**	**Requirements**	**Resp.**	**Source**

The Spec. Sheet

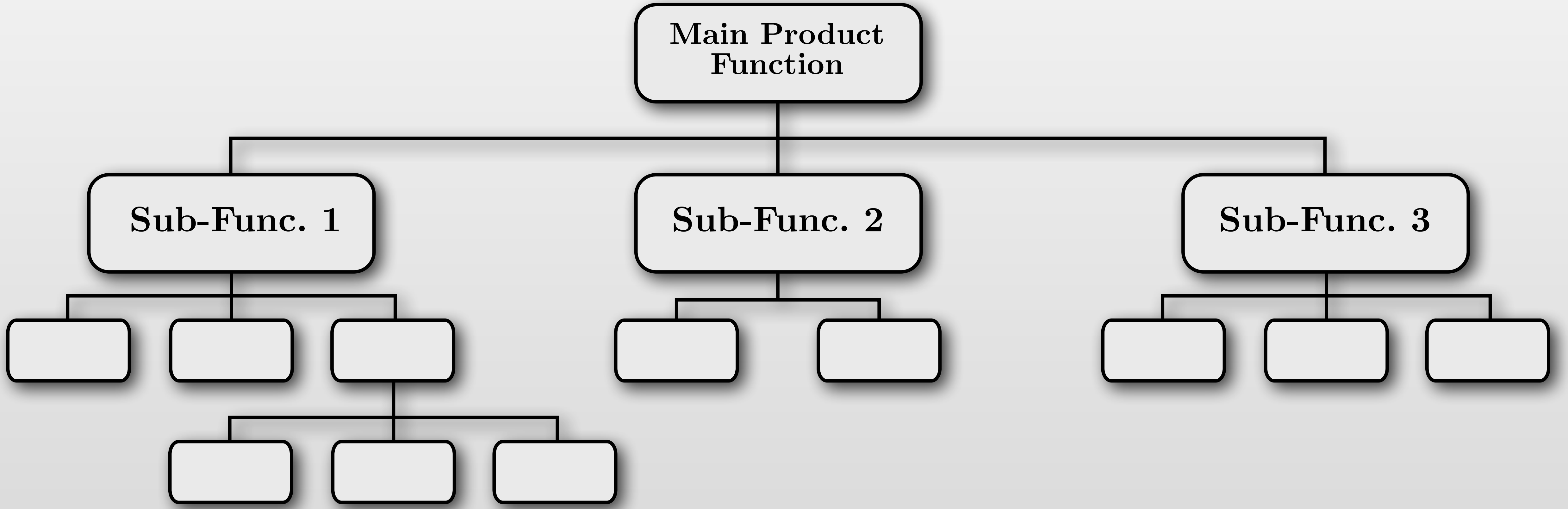
		For: **PRODUCT NAME**	**Issued:** mm/dd/yy Page x of N	
Changes	**D/W**	**Requirements**	**Resp.**	**Source**
Date of last change.	**Demand or Wish?**	**Requirements, sorted by category.**	**Who is responsible?**	**What is the source of this requirement?**

Spec. Sheet Tips/Hints

- Take every Eng. Char. and define numerical targets
  - Number of assembly steps → Number of assembly steps <15
  - Top speed → Top speed > 160mph
  - Max. Acceleration → 0-60mph time <3s
  - Battery life in hours at 75% load → >5 hour life at 75% load
  - "Average" user time between charges → >1 day between charges in normal use
  - Uptime → Uptime > 99.99%
  - Time between required maintenance → Maintenance <1 time per year
- Determine if each is a demand or wish/want
- For every demand, are there accompanying specs. that are wishes/wants?
- Revisit regularly as your understanding of the problem improves!!!

Function Trees

- Break large, difficult design process into many small easy ones



- Continue until the sub-functions are almost trivial

Function Trees (cont.)

- Functions are actions the design must be capable of ***doing***

- Function Tree levels represent complexity, not time ordering (It's not a flow chart.)

- Functions are *NOT*
  - Specific solutions – *e.g.* "Move arm 180 deg."
  - Constraints or specs – *e.g.* "Be smaller than..."

Function Tree Creation Process

- Start by listing the main function of the device
  - Win MCHE201 Final Contest
  - Fold Washed Laundry

- Expand to high-level subfunctions
  - Find Laundry Pile
  - Move to Laundry Pile
  - Fold Each Piece of Laundry
  - …

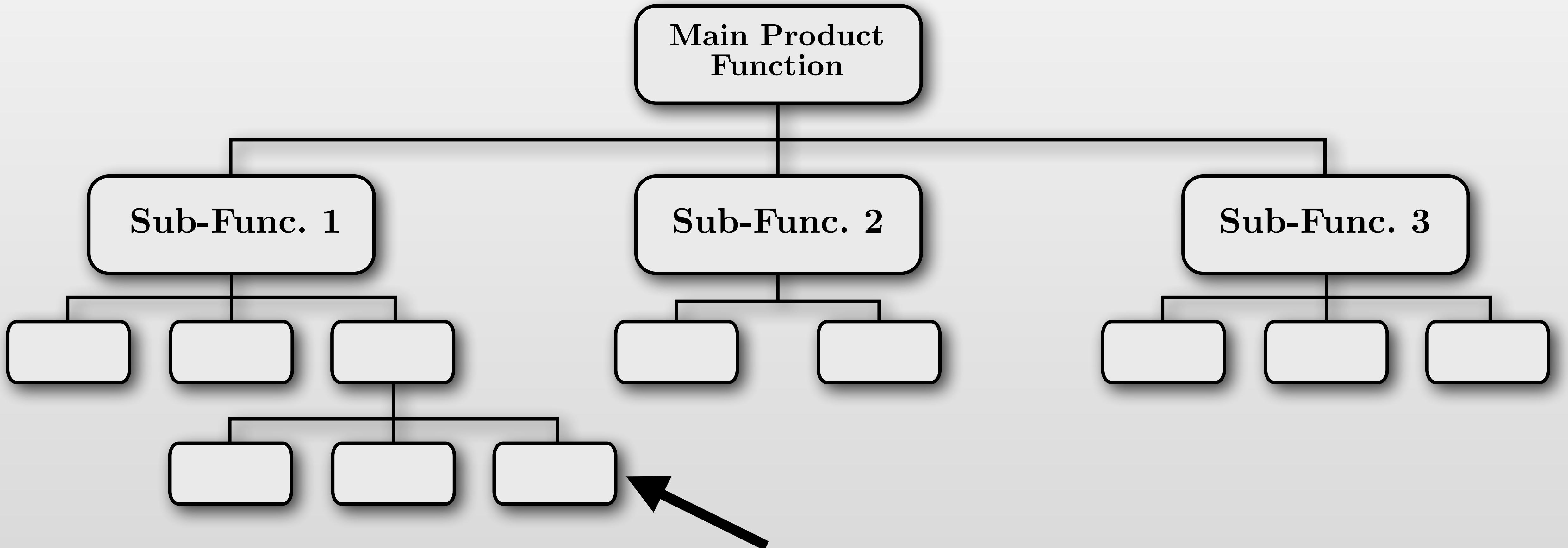
Function Tree Creation Process (cont.)

- Start by listing the main function of the device
- Expand to high-level subfunctions
- For each, list *everything* needed to accomplish that function – For *Fold Each Piece of Laundry*:
  - Recognize Individual Laundry Pieces
  - Calculate Type of Laundry Piece
  - Catalog Total Laundry Pile Characteristics
  - Plan Folding Order for the Pile
  - Move to Individual Laundry Piece
  - Interface with Individual Laundry Piece
  - Calculate Folding Algorithm for Current Piece
  - …

Function Tree Creation Process (cont.)

- Start by listing the main function of the device
- Expand to high-level subfunctions
- For each, list *everything* needed to accomplish that function
- For each new subfunction, list *everything* needed to accomplish that function – For *Move to Individual Laundry Piece*
  - Sense Surrounding Environment
  - Identify Obstacles
  - Identify Orientation of the Laundry Piece
  - Calculate Trajectory to Laundry Piece
  - …

Function Trees



These represent the *complete* set of subfunctions necessary to complete the function immediately above

In-class Exercise

- Develop a Function Tree for a coffee-making robot
  - Takes coffee-shop-like orders
  - Delivers resulting drink to office worker's desk

Morphological Charts

These are from the lowest level of the Function Tree.

	Idea 1	Idea 2	Idea 3	…	…	…	…
Subsubsub func 1							
Subsubsub func 2							
Subsubsub func 3							
...							
...							
...							

Morphological Charts

These are from the lowest level of the Function Tree.

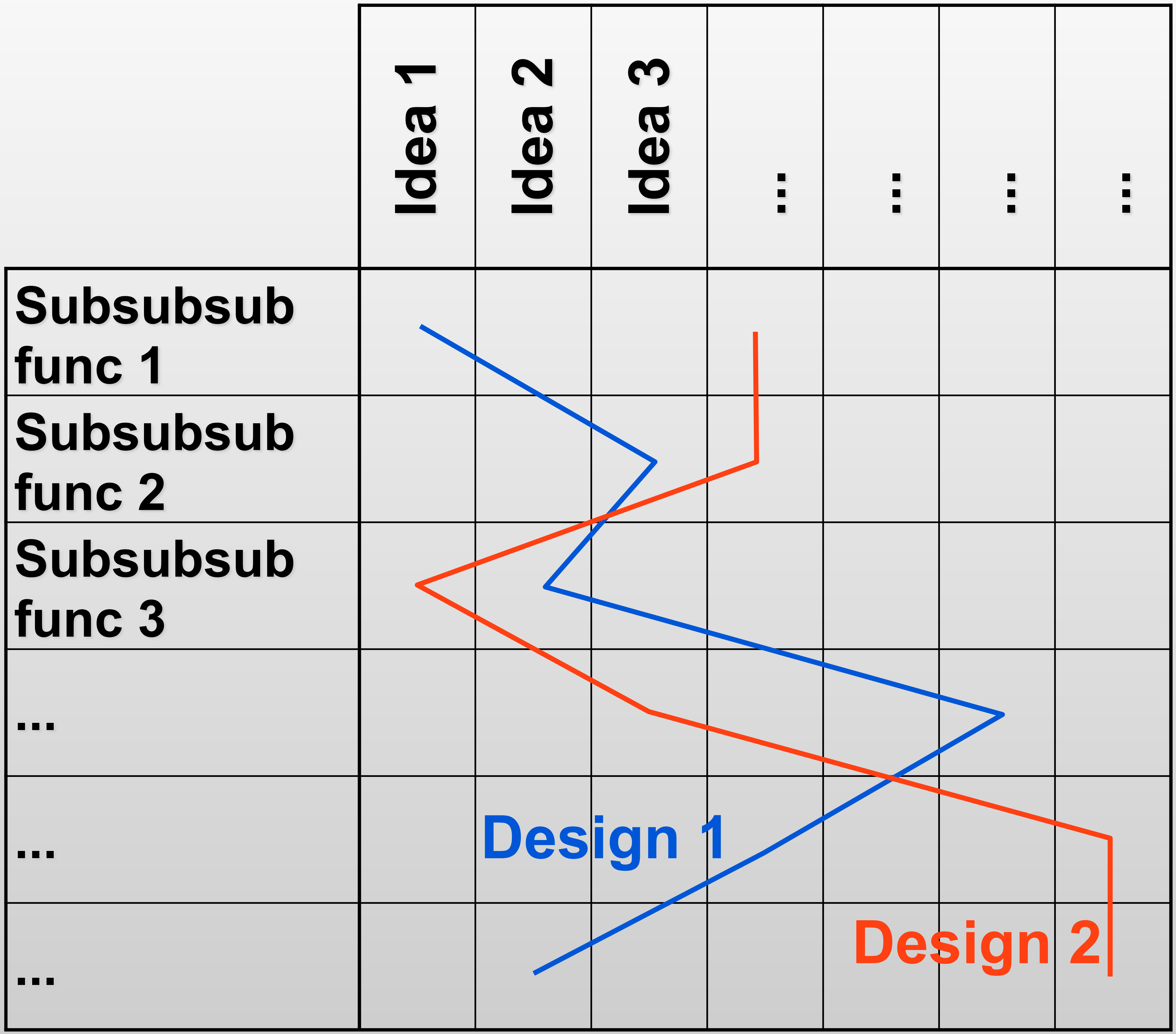
	Idea 1	Idea 2	Idea 3	…	…	…	…
Subsubsub func 1							
Subsubsub func 2							
Subsubsub func 3							
…							
…							
…							

Each cell is a sketch.

Simple representation of the idea.

Allows for quick, visual comparisons.

Morphological Charts



Alternative Concepts

- All must satisfy same
  - Customer Req.
  - Specifications
  - Functions

- Want many *unique* designs that are all great

  (Such that choosing the *final* one is difficult)

How should we choose?

The Problem Understanding Form

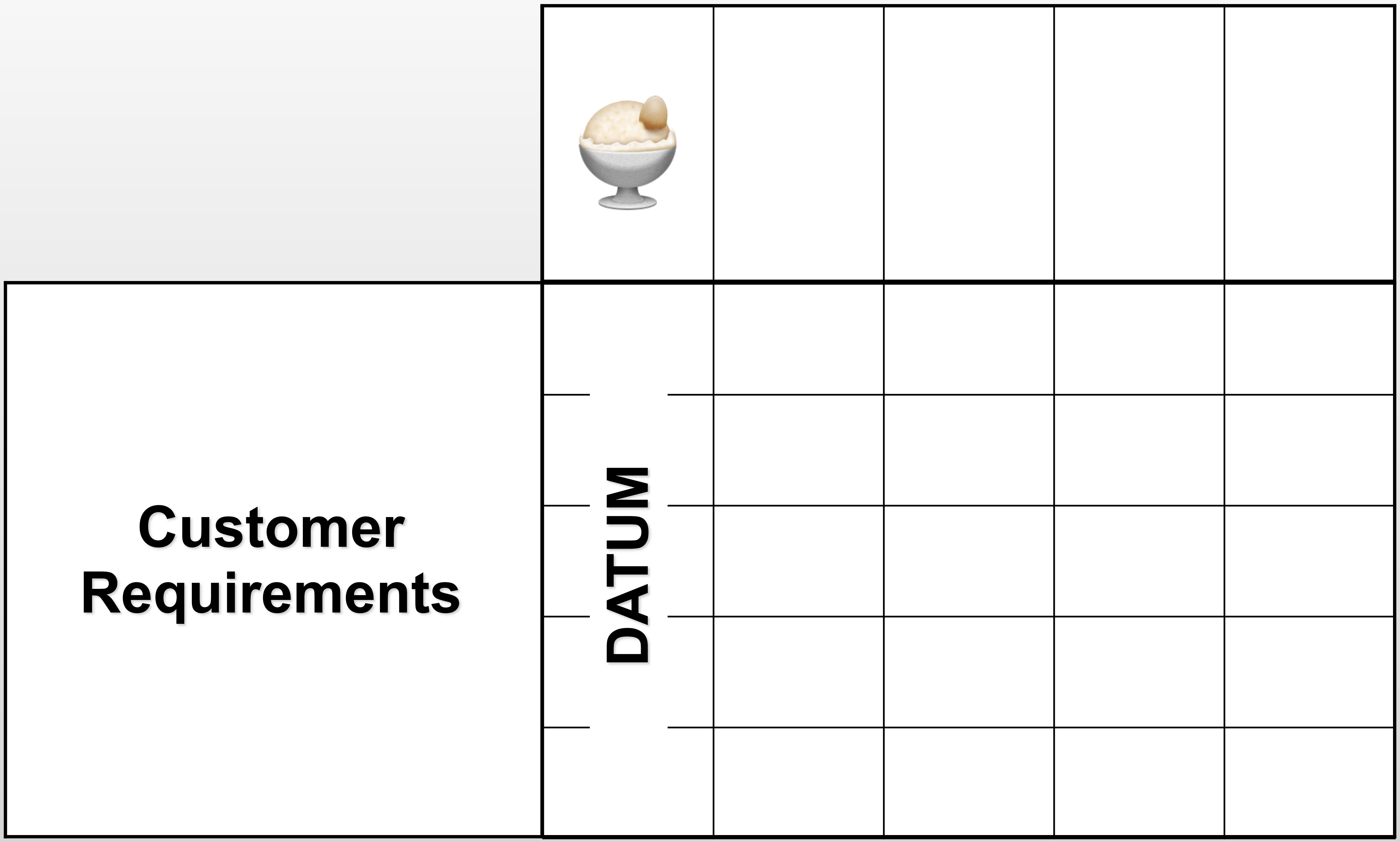
◉	Strong = 9
▣	Medium = 3
△	Weak = 1

Customer Requirements		Engineering Characteristics				
	5	◉		▣	△	
	6		◉	▣		▣
	9	◉	▣		△	
	2	▣	△			▣
	1		◉	△	◉	△
Absolute Importance		132	92	34	23	73
Relative Importance		0.37	0.26	0.1	0.06	0.21

1st-Level Evaluation Matrix

	Alternative Concepts
Customer Requirements	Evaluation of Concept Against Cust. Req.

1st-Level Eval Matrix - Dessert



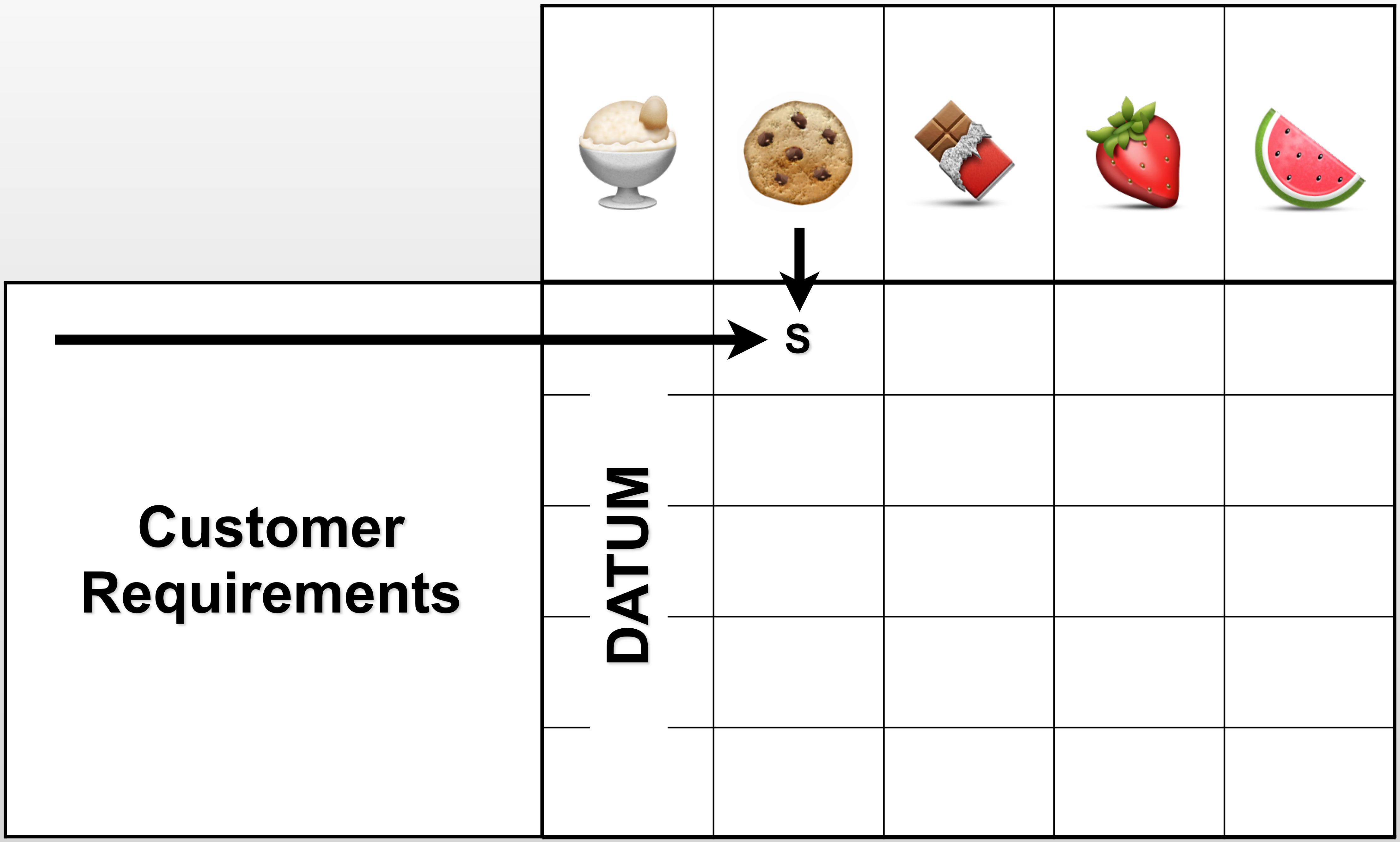
- Choose a “benchmark” product as your datum

1st-Level Eval Matrix - Dessert

					
Customer Requirements	DATUM				

- Fill in your concepts and compare to datum
- + = better, - = worse, S = same

1st-Level Eval Matrix - Dessert



- Fill in your concepts and compare to datum
- + = better, - = worse, S = same

1st-Level Eval Matrix - Dessert

					
Customer Requirements	DATUM	S	S		

- Fill in your concepts and compare to datum
- + = better, - = worse, S = same

1st-Level Eval Matrix - Dessert

					
Customer Requirements	**DATUM**	S	S	–	–
		+	+	+	S
		–	–	+	S
		S	S	+	+
		–	+	S	–

- Fill in your concepts and compare to datum
- + = better, - = worse, S = same

1st-Level Eval Matrix - Dessert

					
Customer Requirements	**DATUM**	S	S	–	–
		+	+	+	S
		–	–	+	S
		S	S	+	+
		–	+	S	–
$\sum +$		1	2	3	1
$\sum -$		2	1	1	2
$\sum S$		2	2	1	2

1st-Level Eval Matrix Weaknesses?

- No consideration of importance

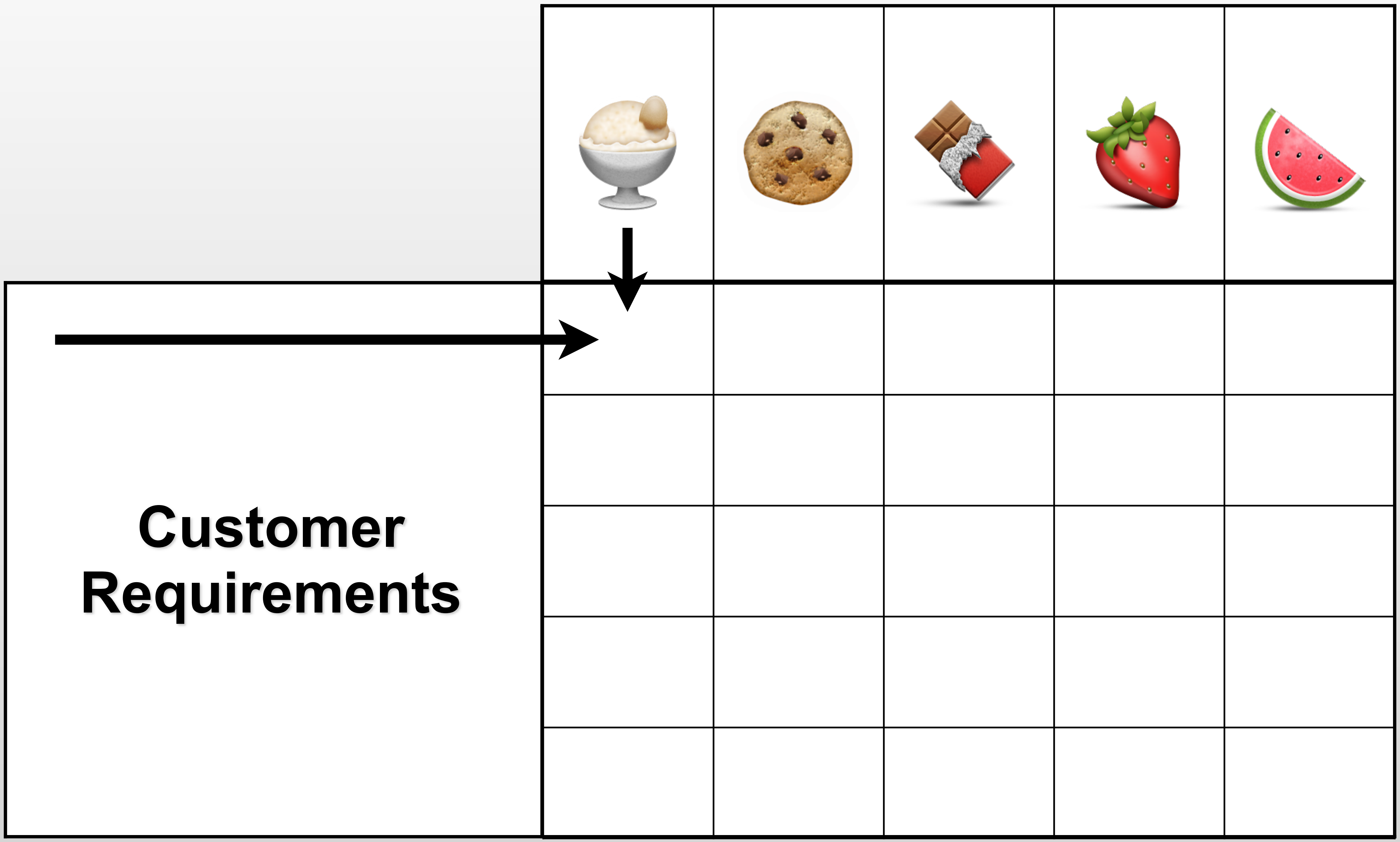
- No indication of *magnitude* of better/worse

2nd-Level Evaluation Matrix

					
Customer Requirements					

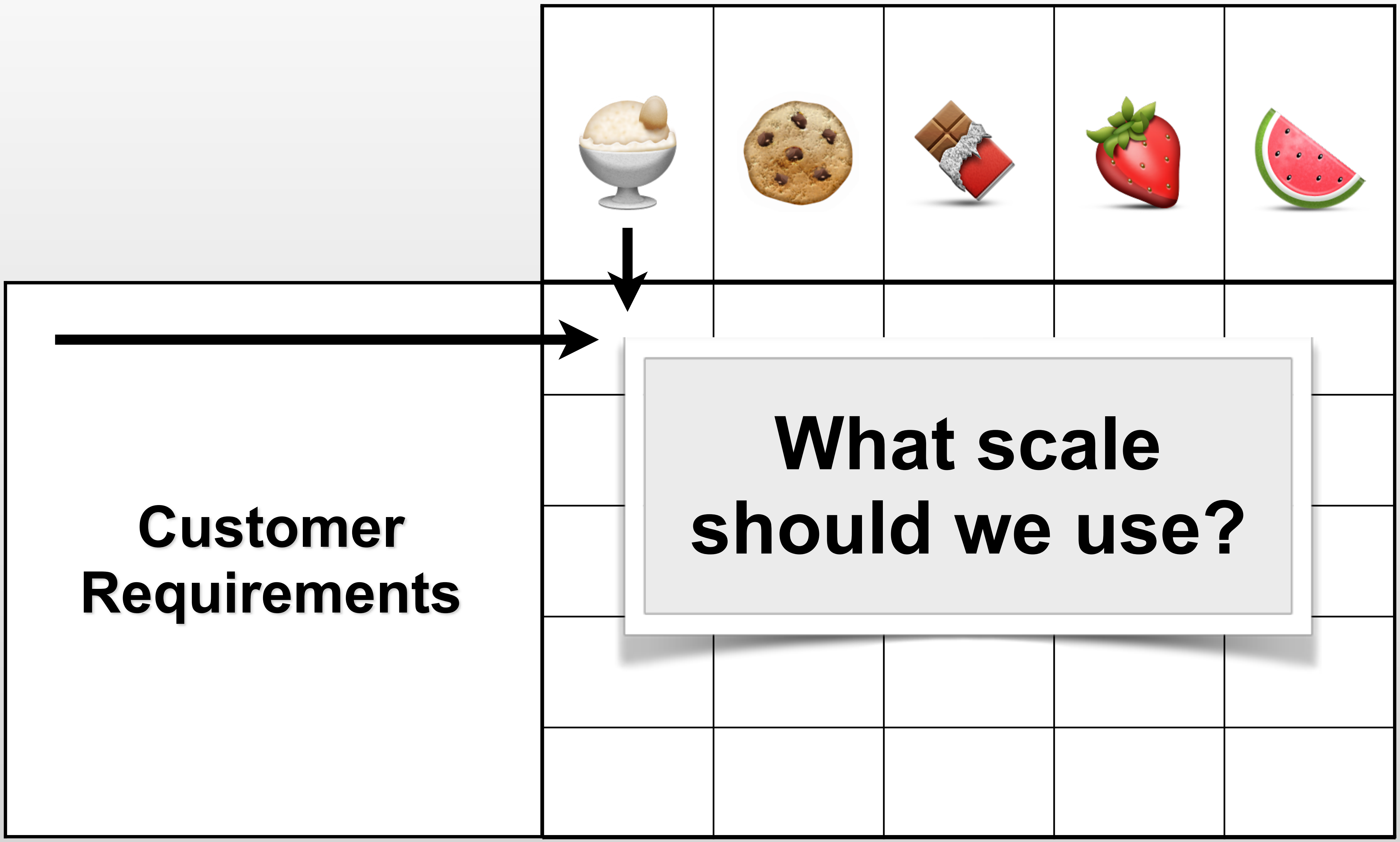
- Give numerical value to how well a Customer Req. is satisfied

2nd-Level Evaluation Matrix



- Give numerical value to how well a Customer Req. is satisfied

2nd-Level Evaluation Matrix



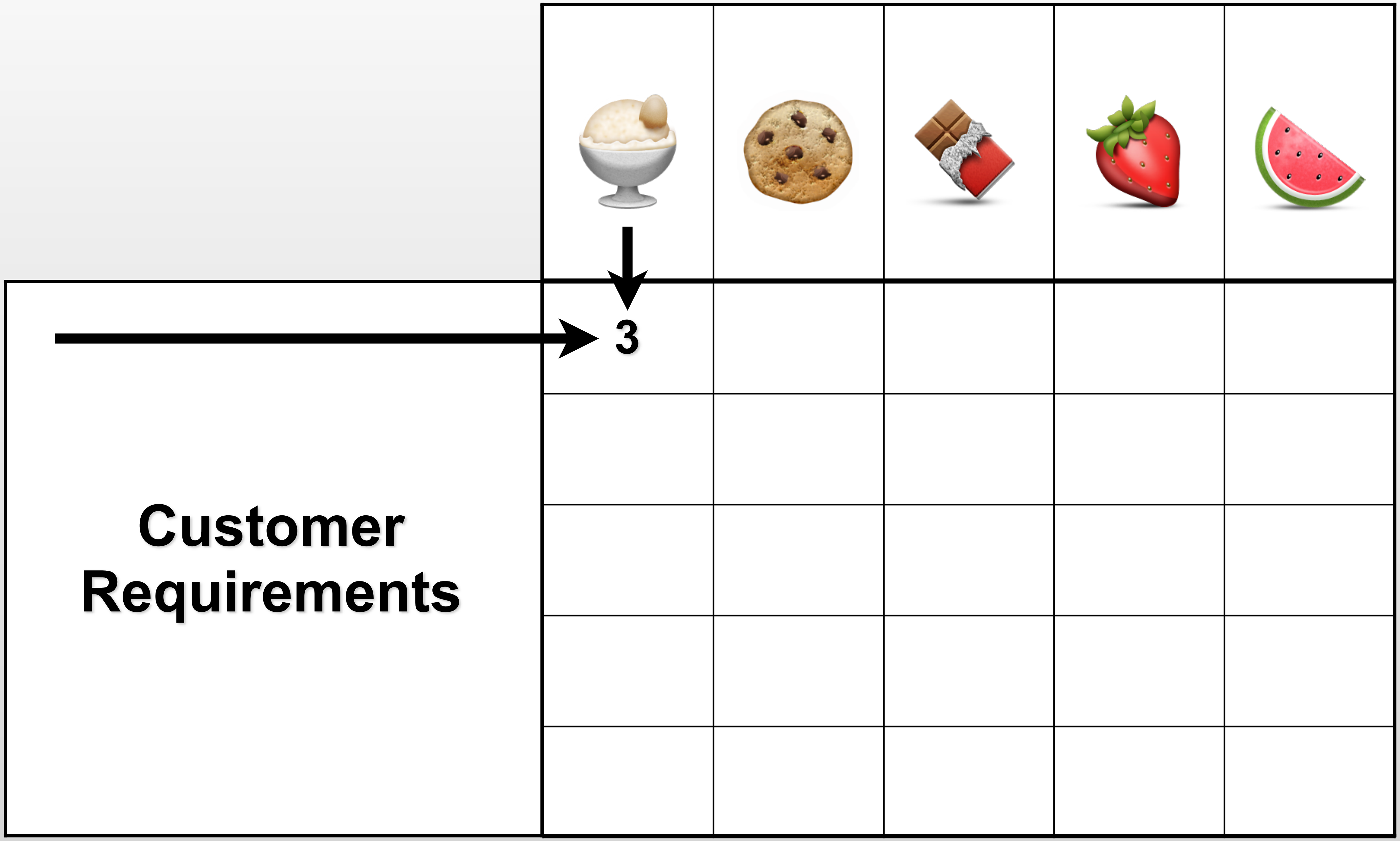
- Give numerical value to how well a Customer Req. is satisfied

Evaluation Scales

Pts	Meaning
0	Unsatisfactory
1	Just Tolerable
2	Adequate
3	Good
4	Very Good (Ideal)

Pts	Meaning
0	Unsatisfactory
1	Inadequate
2	Weak
3	Tolerable
4	Adequate
5	Satisfactory
6	Good, but drawbacks
7	Good
8	Very Good
9	Exceeds Req.
10	Ideal Solution

2nd-Level Evaluation Matrix



2nd-Level Evaluation Matrix

					
Customer Requirements	3	3	4	2	2
	0	3	4	2	1
	1	1	1	4	3
	2	2	2	3	4
	2	4	1	0	1

2nd-Level Evaluation Matrix

					
Customer Requirements	3	3	4	2	2
	0	3	4	2	1
	1	1	1	4	3
	2	2	2	3	4
	2	4	1	0	1
Absolute Total	8	13	12	11	11

2nd-Level Evaluation Matrix

					
Customer Re	3	3	4	2	2
	0	3	4	2	1
Absolute Total	8	13	12	11	11

$$\text{Relative Total} = \frac{\text{Absolute Total}}{\text{Max. Possible}}$$

2nd-Level Evaluation Matrix

					
Customer Requirements	3	3	4	2	2
	0	3	4	2	1
	1	1	1	4	3
	2	2	2	3	4
	2	4	1	0	1
Absolute Total	8	13	12	11	11
Relative Total	0.4	0.65	0.6	0.55	0.55

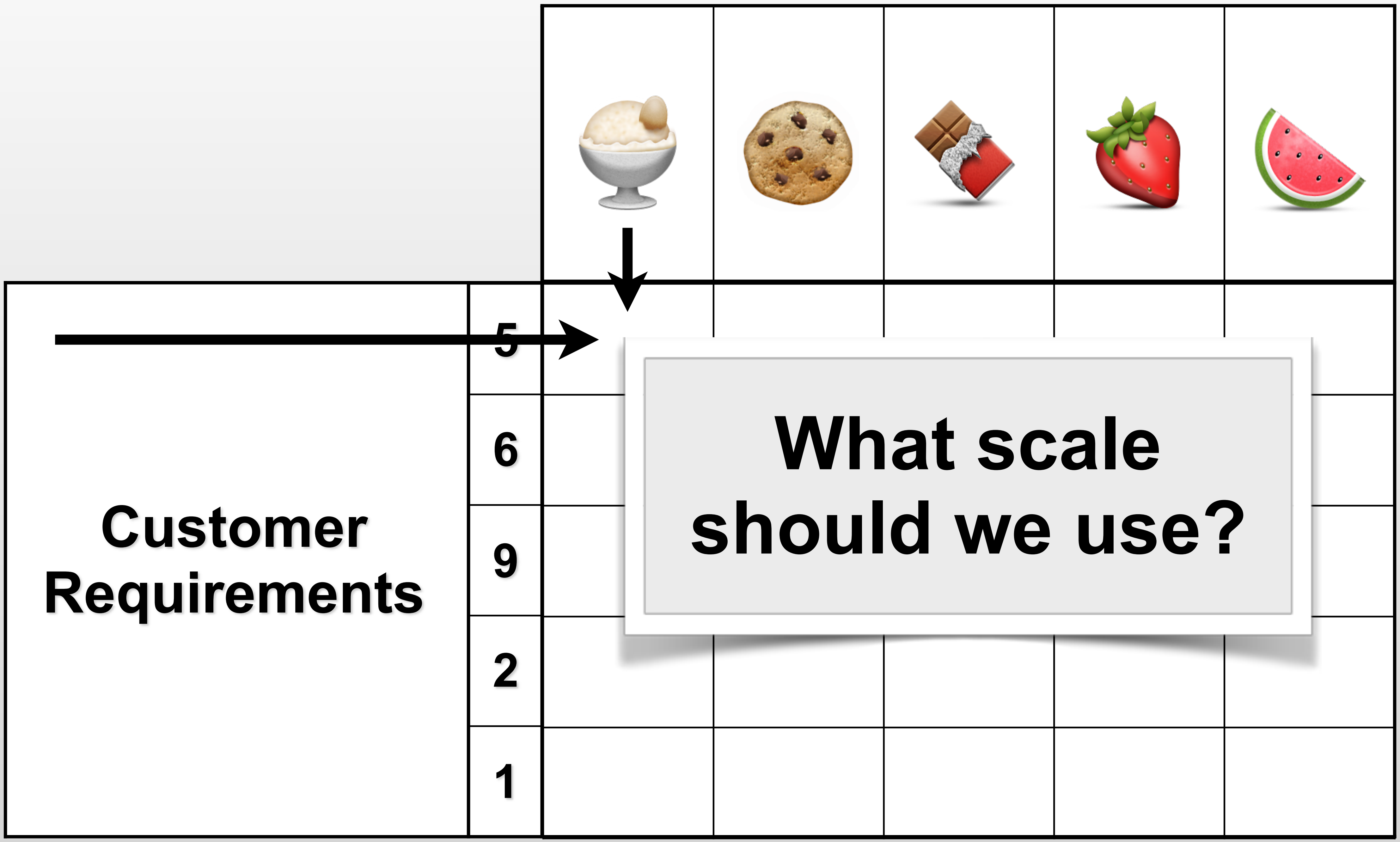
2nd-Level Eval Matrix Weaknesses?

- No consideration of importance

3rd-Level Evaluation Matrix

						
Customer Requirements	5					
	6					
	9					
	2					
	1					

3rd-Level Evaluation Matrix



Evaluation Scales

Pts	Meaning
0	Unsatisfactory
1	Just Tolerable
2	Adequate
3	Good
4	Very Good (Ideal)

Pts	Meaning
0	Unsatisfactory
1	Inadequate
2	Weak
3	Tolerable
4	Adequate
5	Satisfactory
6	Good, but drawbacks
7	Good
8	Very Good
9	Exceeds Req.
10	Ideal Solution

3rd-Level Evaluation Matrix

Customer Requirements						
	5	7				
	6					
	9					
	2					
	1					

3rd-Level Evaluation Matrix

						
Customer Requirements	5	7	6	9	5	4
	6	0	7	10	5	2
	9	3	2	3	10	8
	2	5	6	5	8	0
	1	6	9	2	0	3

3rd-Level Evaluation Matrix

						
Customer Requirements	5	7	6	9	5	4
	6	0	7	10	5	2
	9	3	2	3	10	8
	2	5	6	5	8	0
	1	6	9	2	0	3
Absolute Total						

Design “Scores”

- Absolute Total

$$\text{Absolute Total} = \sum_{\text{col}} (\text{Design Performance} \times \text{Customer Importance})$$

- Relative Total

$$\text{Relative Total} = \frac{\text{Absolute Total}}{\text{Max. Possible}}$$

3rd-Level Evaluation Matrix

		🍨	🍪	🍫	🍓	🍉
Customer Requirements	5	7	6	9	5	4
	6	0	7	10	5	2
	9	3	2	3	10	8
	2	5	6	5	8	0
	1	6	9	2	0	3
Absolute Total		78	111	134	161	107
Relative Total		0.34	0.48	0.58	0.7	0.47

Remember that...

- Much of the utility of these is in having to think objectively about the designs to rate them

- A design is the not the best because it got the highest score. It got the highest score because it's the best.

- Like all the tools, these are "living" documents
  - They are part of the design process, not the end of it
  - Can identify weaknesses in otherwise good designs
  - Promotes "cross pollination" of ideas
  - *ITERATE!*