Requirements Specification

ECEE Senior Design

The Bigger Picture

- After you finalize Requirements Specification

  Need to get from system requirements to …

  High-level elements (RRC) organized in a *System Diagram* showing

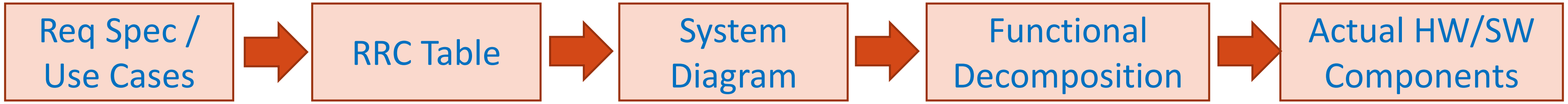
  - Abstract hardware elements (**not** part #’s yet)
  - Abstract software elements
  - Nature of communication between them (what data, signal, …)

  …that can accomplish the requirements, including Use Cases

- Then go from System Diagram to…

  Functional Decomposition of hardware

  Software architecture – UML class diagram



Benefits of a Good Requirements Spec.

- Establish the basis for agreement between the customers and the suppliers on what the product is to do
- Reduce the development effort
- Provide a basis for estimating costs and schedules
- Provide a baseline for validation and verification
- **Facilitate product transfer to new users, machines, business units, customers, etc.**
- Serve as a basis for enhancement

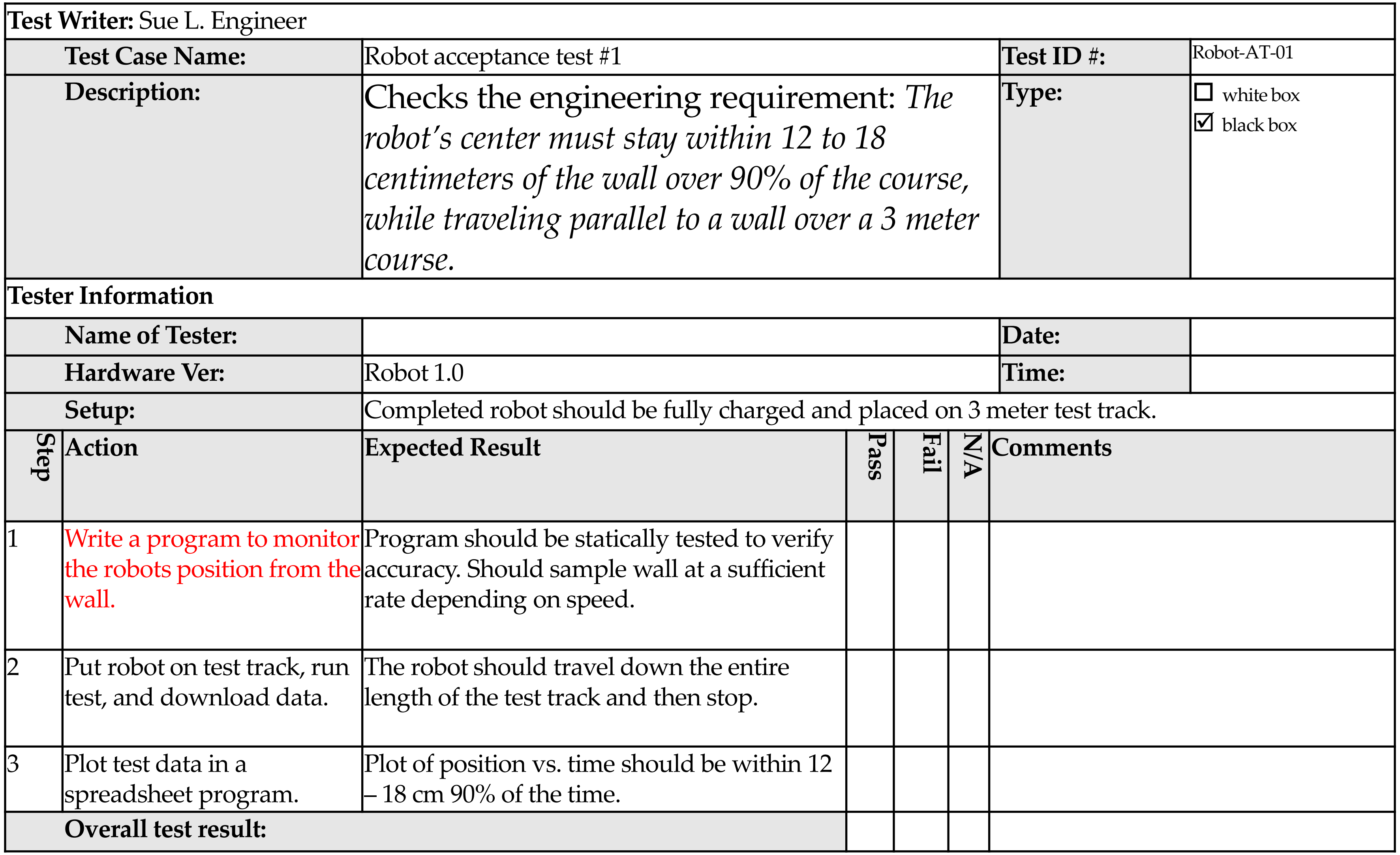
IEEE Std 830-1998, Introduction

What is a requirement?

- An externally visible function or attribute of a system
- Requirements Spec addresses the ***product***, not the process of producing it
- Items such as cost, schedule, methods, tests do not appear here

- Your book – Chapter 3
  - Marketing requirements
    - Short statements describe needs from the user's perspective
  - Engineering requirements
    - Short statements that address a technical need of the design
      - Some are derived from marketing requirements
      - Some are in addition to marketing requirements
- Your text – Chapter 7 – HIGHLY RECOMMEND YOU READ AHEAD
  - 7.2.4 Acceptance Test
  - 7.3 Case Study

Robot Acceptance Test

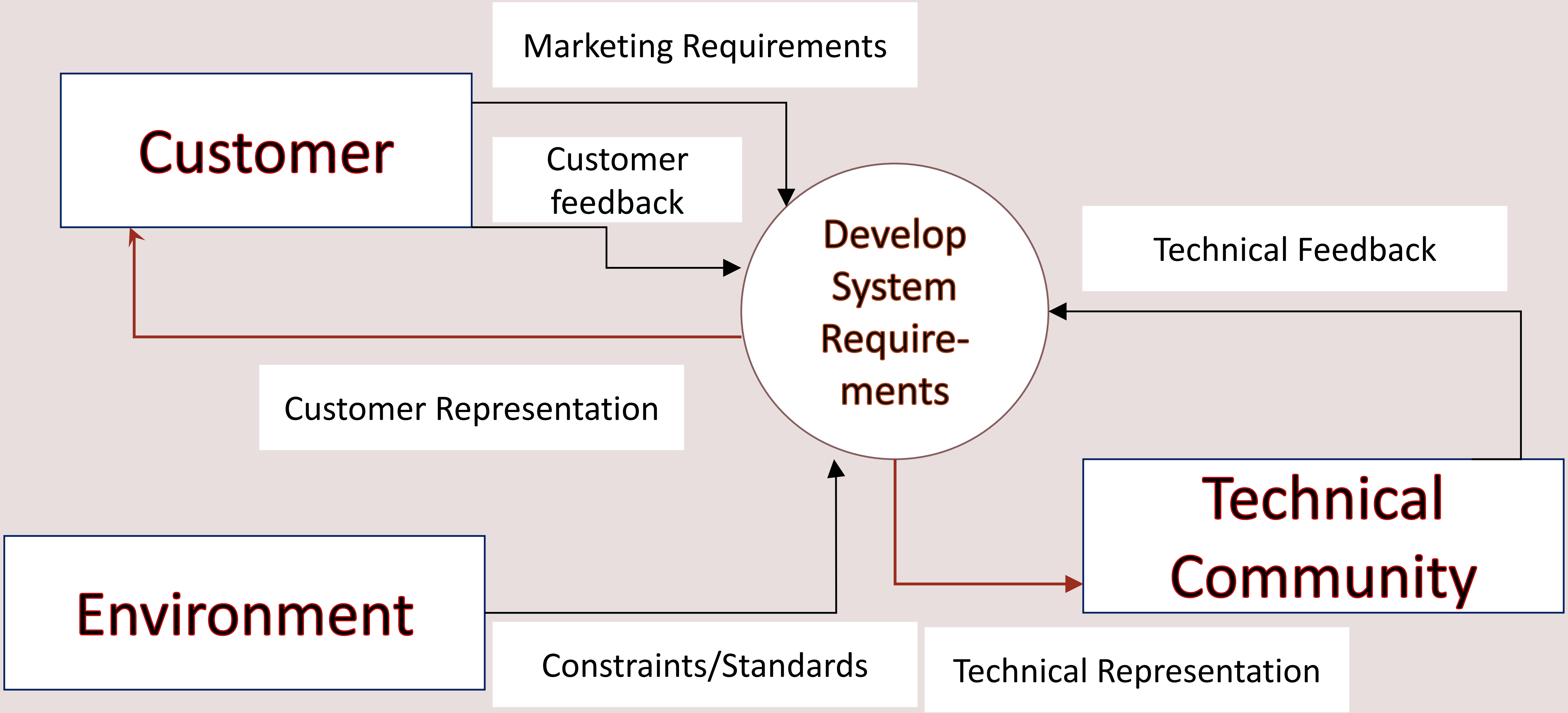


Test Writer: Sue L. Engineer						
	Test Case Name:	Robot acceptance test #1				**Test ID #:** Robot-AT-01
	Description:	Checks the engineering requirement: *The robot's center must stay within 12 to 18 centimeters of the wall over 90% of the course, while traveling parallel to a wall over a 3 meter course.*				**Type:** ☐ white box ☑ black box
Tester Information						
	Name of Tester:					**Date:**
	Hardware Ver:	Robot 1.0				**Time:**
	Setup:	Completed robot should be fully charged and placed on 3 meter test track.				
Step	**Action**	**Expected Result**	**Pass**	**Fail**	**N/A**	**Comments**
1	Write a program to monitor the robots position from the wall.	Program should be statically tested to verify accuracy. Should sample wall at a sufficient rate depending on speed.				
2	Put robot on test track, run test, and download data.	The robot should travel down the entire length of the test track and then stop.				
3	Plot test data in a spreadsheet program.	Plot of position vs. time should be within 12 – 18 cm 90% of the time.				
	Overall test result:					

NO.

Acceptance Test should be a direct extension of your Requirements

Fig 3.1 from text, taken from IEEE *Guide for Developing System Requirements Specifications*, IEEE Std 1233-1998



Customer

Marketing Requirements

Customer feedback

Develop System Requirements

Technical Feedback

Customer Representation

Environment

Constraints/Standards

Technical Community

Technical Representation

Characteristics of well-formed Engineering Requirements

- *Abstract – what* does the system do – not *how*
  - *G*ive yourself implementation freedom!
  - Think about the “bridge” example
- *Verifiable* or *Can be demonstrated* that it does X (measurable)
- *Unambiguous* – everyone agrees on it
- *Traceable*
  - What marketing requirement prompted this, if any?
  - What is the <u>rationale</u> for it?
  - WHO is requiring it?

How do I think of all the requirements?

Checklist pp 40-49 is a good place to start

- **System interfaces** (not component interfaces)
- **Functionality** at the **system level**
- **Quality Attributes** of the system
  - Performance
  - Reliability/Availability
  - Portability
  - Other –ilities

Additional Categories to consider

- Economic requirements – e.g. cost per unit
- Energy consumption; energy usage
- Environmental impact: Radiation? Human safety? Chemical output?
- Legal: no IP infringement?
- Maintainability
- Manufacturability
- Operational conditions (temperature, moisture, vibrations, withstand drop, etc.)
- Governmental
- Social/cultural/ADA
- Usability

Format based on IEEE Standard 830-1998

- **1. Introduction**
- 1.1 Purpose -- intended audience
- 1.2 Scope – what product is being described here; what problem is it solving
- 1.3 Definitions, acronyms, and abbreviations -- don't define what TAs and instructors will know
- 1.4 References -- if you reference a document elsewhere in this spec, list it here, too.
- 1.5 Overview (high level view of key points in this documents)
- **2. Overall description**
- 2.1 Product perspective – simple block diagram of system components
- 2.2 Product functions -- summarize in *priority order* (indicate intended cutoff for project)
- 2.3 User characteristics or Larger system context characteristics
- 2.4 Design Constraints
- 2.5 Assumptions and dependencies
- **3.0 Specific Requirements [See Chapter 3 in course text]**
- **4.0 Use Cases [Extra reference provided on D2L]**

2.0 Overall description

2.1 Product perspective --

- Block diagram of ***major*** components, interconnections, and external interfaces

and/or

- How the system operates under various constraints which may include
  - system interfaces – what interfaces in the larger system is your product required to meet?
  - user interfaces -- *logical* characteristics of each interface the system presents to the user; required display layouts; forms; reports; indicators; constraints due to user characteristics
  - hardware interfaces -- *logical* characteristics of each interface between the major software and hardware components of the system such as number of ports and their purposes, what devices will be supported for what purpose.
  - (continued on next slide)

2.1 Product perspective (continued)

- software interfaces -- use of other required software products (are you using some open-source tool such as a data base app, math library, operating system?) Available reference material.

- communications interfaces -- various interfaces to communications such as local network protocols

- memory -- relevant characteristics or limits on primary and secondary memory (on-board memory? flash drive available?)

- operations -- normal and special operations required by user such as
  - modes of operation (e.g., novice/expert; interactive/autonomous; etc.)
  - backup/recovery operations
- site adaptation requirements
  - requirements for data or initialization sequences specific to a given site, mission, or operational mode (safety limits, ...)
  - anything that must be done to adapt the product to a particular installation

2.2 Product Functions Summary

- High: Without these features we don’t have a product
- Medium: The customer/sponsor would really like to have this
- Low:
  - Cool to have but prototype demo will be great without it anyway
  - It’s the next likely feature the customer would want
  - It’s an interesting extension for an independent study

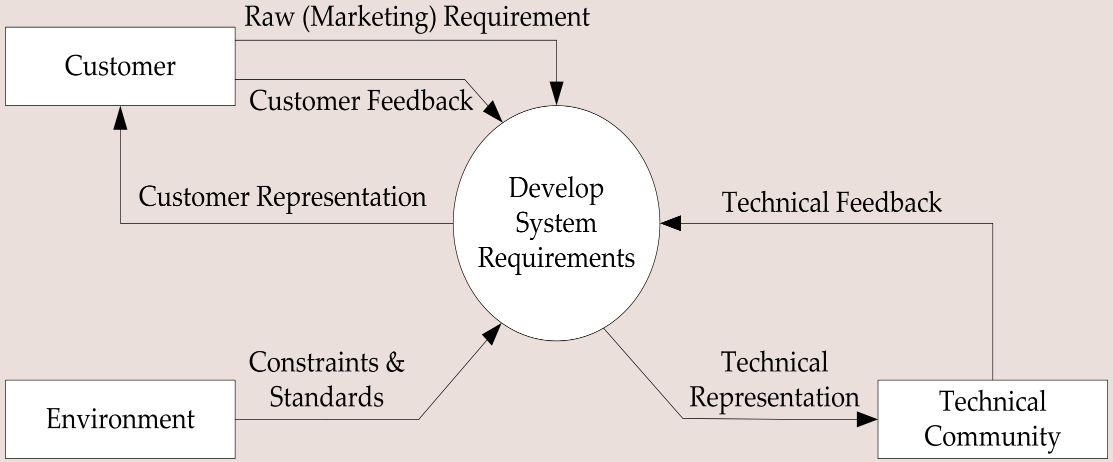
2.3 User Characteristics or Larger System Characteristics

- User characteristics in its intended working context
  - Expectation of technology comfort level?
  - Educational level?
  - Knowledge in a particular area?
  - Experts in their own domain? Or beginners?
  - Particular disabilities to accommodate?
  - Language challenges?
  - Outdoors?
- If your box goes inside someone else's larger system, *that* is the user.
- Larger System Characteristics
  - In outer space?
  - Outdoors?
  - Dimensional constraints?
  - ???

2.4 Design Constraints

These are NOT the same as Marketing Requirements, which still leave implementation freedom.

- Anything that limits your options as developers
  - Regulatory policies
  - Hardware limitations
  - Interfaces to other applications
  - Audit functions
  - Control functions
  - Must write in C++ ?? (just an example)
  - DON’T use this to capture your design ideas that are your choice and not imposed on you



2.5 Assumptions and Dependencies

- Are you assuming … or … who/what are you dependent on (but not in control of)?
    - The XYZ Space Grant team has submodule ABC available by November 1 for us to use with our prototypes?
- Focus on non-obvious, unique and specific items

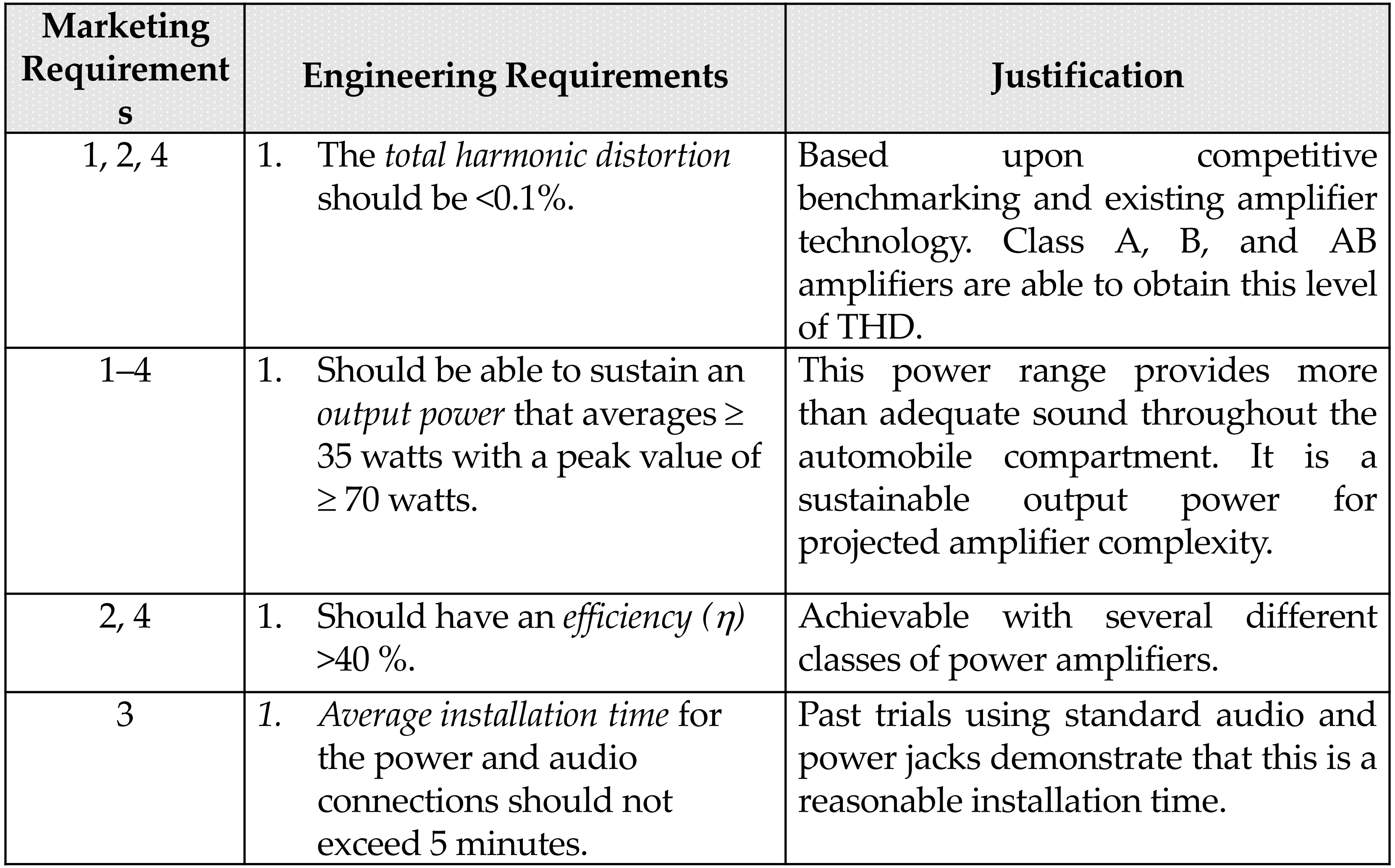
3.0 Specific Requirements

- Book shows a table that looks like this:

Marketing requirements	Engineering Requirements	Justification/Rationale
Refer by number for traceability	Derived requirements	

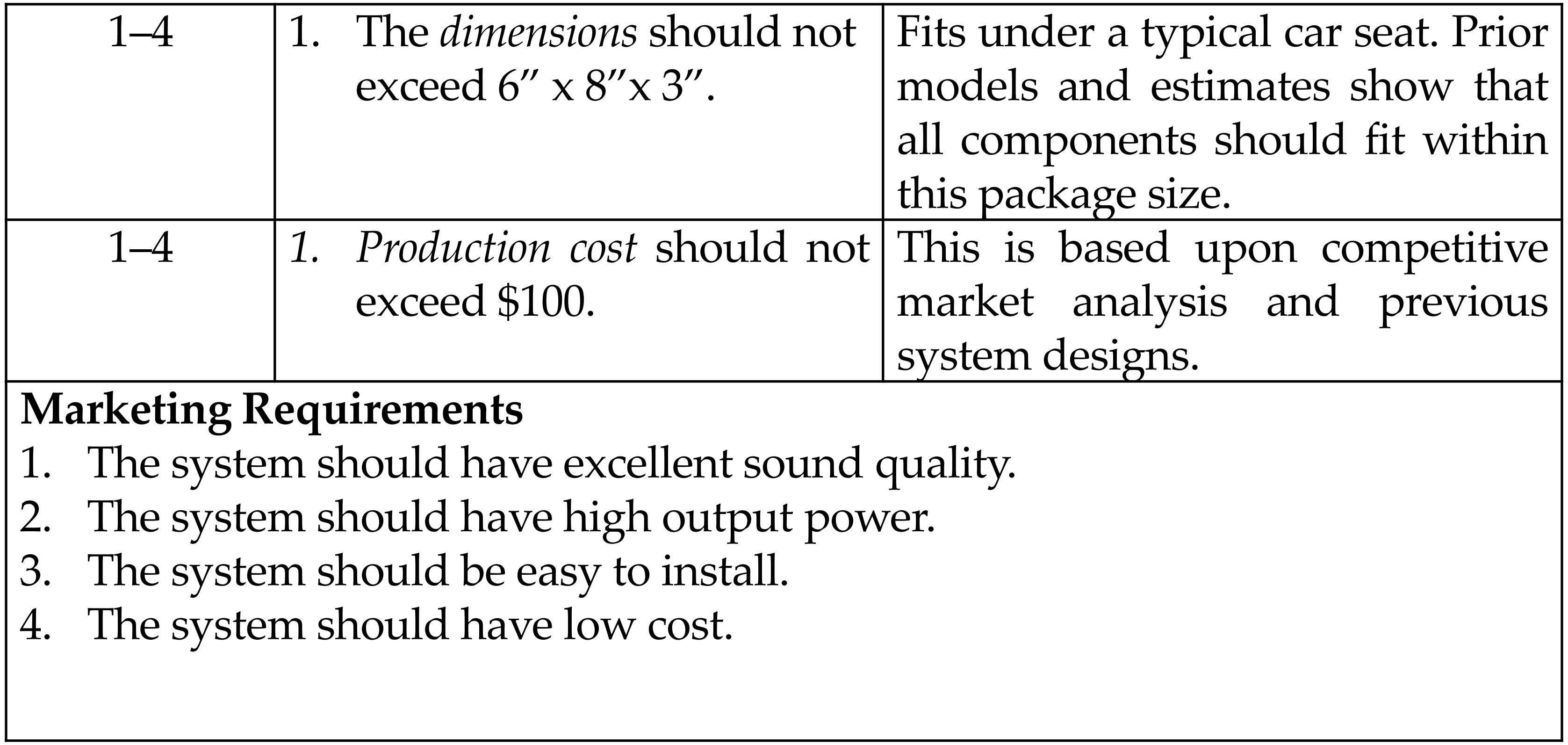
Marketing Requirements Summary
1.
2.
3.
n.

3.4 Case Study: Car Audio Amp



Marketing Requirement s	Engineering Requirements	Justification
1, 2, 4	1. The *total harmonic distortion* should be <0.1%.	Based upon competitive benchmarking and existing amplifier technology. Class A, B, and AB amplifiers are able to obtain this level of THD.
1–4	1. Should be able to sustain an *output power* that averages ≥ 35 watts with a peak value of ≥ 70 watts.	This power range provides more than adequate sound throughout the automobile compartment. It is a sustainable output power for projected amplifier complexity.
2, 4	1. Should have an *efficiency* (*η*) >40 %.	Achievable with several different classes of power amplifiers.
3	*1. Average installation time* for the power and audio connections should not exceed 5 minutes.	Past trials using standard audio and power jacks demonstrate that this is a reasonable installation time.

Case Study, cont’d



1–4	1. The *dimensions* should not exceed 6″ x 8″x 3″.	Fits under a typical car seat. Prior models and estimates show that all components should fit within this package size.
1–4	*1. Production cost* should not exceed $100.	This is based upon competitive market analysis and previous system designs.
Marketing Requirements 1. The system should have excellent sound quality. 2. The system should have high output power. 3. The system should be easy to install. 4. The system should have low cost.		

How do you VALIDATE requirements?

- Ask the customer if the requirements meet their needs
- Usually done in teams
- For each *engineering requirement* [Try assigning a team member to this!]
    - Abstract? (what, *not* how system works)
    - Verifiable? (measurable, repeatable)
    - Unambiguous? (clear enough meaning to agree on, not the details of how)
    - Traceable? (tied to marketing requirement, true customer need)
- For the complete *Requirements Specification*
    - Orthogonal?
        - No overlap or redundancy between engineering requirements
    - Complete?
        - Addresses all needs of end user and those to implement the system
    - Consistent?
        - Not self-contradictory
    - Bounded
        - Scope of requirements specification identified
    - Modifiable
        - Estimates provided for baseline requirements in rev 1.0, then solidified for 2.0 or even 3.0 by CDR

4.0 Use Cases (not part of IEEE std 830-1998, nor in text)

- Describe interaction with product from “user” or external system perspective
- **Scope**
- **Level**
- **Primary actor** (person/role, another system, who initiates)
- **Stakeholders**
- **Preconditions** – what *must* be true when this user case begins
- **Postconditions** – success guarantee. If preconditions are met and this use case occurs, what outcome is guaranteed by the system?
- **Main success scenario (Basic flow)**
- **Extensions (Alternative Flows)**

4.0 Use Cases (continued)

- **Special Requirements**
  - Response time?
  - Display must be visible from 12 feet
  - ???
- **Technology and Data Variations List**
- **Frequency of occurrence of this use case**
- **Open Issues** (stuff you don’t now yet but you think you need to know)

In summary…
YOUR Requirements Specification for ECEN-4610

- **Based on IEEE Standard 830-1998, which is centered on software, but we are generalizing for system (both hardware and software)**
- **Follow the format and guidelines provided in ECEN-4610 ReqSpecOutline.doc.**
- **Additional reference for sections 1 and 2 provided in IEEE Spec 830-1998.**
- **Section 3 should follow the format, examples and explanation in Chapter 3 of your course text.**
- **Section 4 should include your Use Cases.**
  - This is not part of the IEEE standard.
  - Reference WhatUseCaseSectionsMean.pdf and UseCaseFullyDressedExample.pdf, which are excerpts from *Applying UML and Patterns*, 3rd Edition, by Craig Larman.
- **Requirements Specification examples from previous Sr Design Teams provided as well**
  - These are not perfect nor ideal! **Use your own cognitive and creative talent.**

Format based on IEEE Standard 830-1998

- **1. Introduction**
- 1.1 Purpose -- intended audience
- 1.2 Scope – what product is being described here; what problem is it solving
- 1.3 Definitions, acronyms, and abbreviations -- don't define what TAs and instructors will know
- 1.4 References -- if you reference a document elsewhere in this spec, list it here, too.
- 1.5 Overview (high level view of key points in this documents)
- **2. Overall description**
- 2.1 Product perspective – simple block diagram of system components
- 2.2 Product functions -- summarize in *priority order* (indicate intended cutoff for project)
- 2.3 User characteristics or Larger system context characteristics
- 2.4 Design Constraints
- 2.5 Assumptions and dependencies
- **3.0 Specific Requirements [See Chapter 3 in course text]**
- **4.0 Use Cases [Extra reference provided on D2L]**

The Bigger Picture

- After you finalize Requirements Specification

  Need to get from system requirements to …

  High-level Components (RRC) organized in a *System Diagram* showing

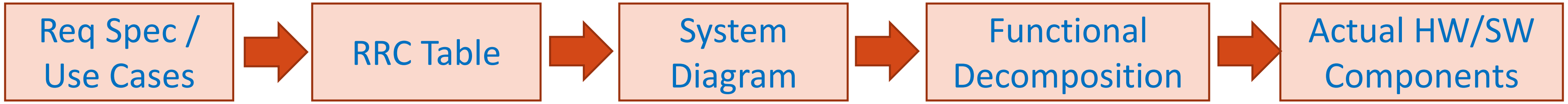
  - Abstract hardware components (**not** part #’s yet)
  - Abstract software components
  - Nature of communication between them (what data, signal, …)

  …that can accomplish the requirements, including Use Cases

- Then go from System Diagram to…

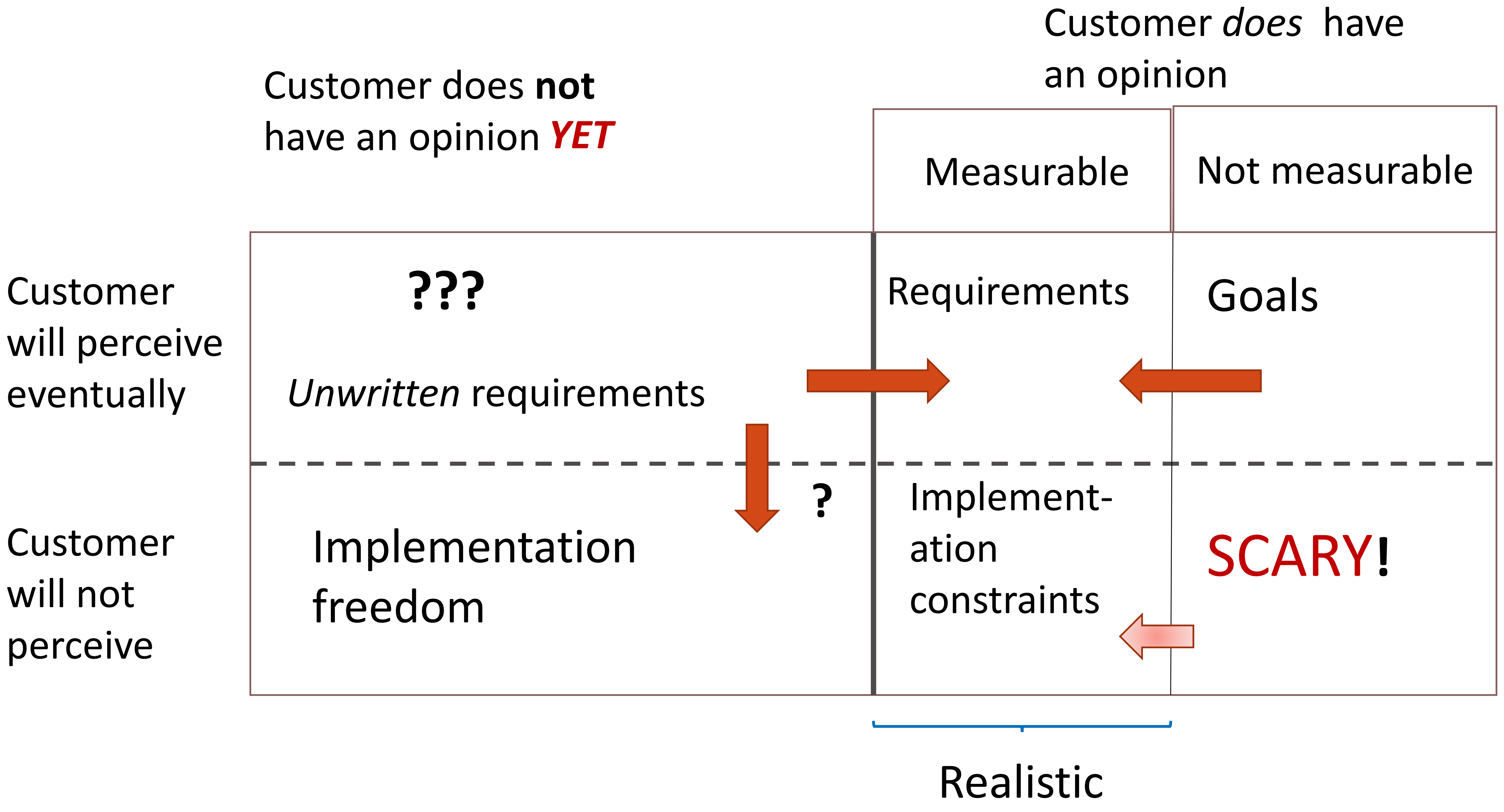
  Functional Decomposition of hardware

  Software architecture – UML class diagram



Reference

What is a requirement? (per David Lamb, Univ of Waterloo)



	Customer does **not** have an opinion ***YET***	Customer *does* have an opinion	
		Measurable	Not measurable
Customer will perceive eventually	**???** *Unwritten* requirements	Requirements	Goals
Customer will not perceive	Implementation freedom **?**	Implementation constraints	SCARY**!**

Realistic