Engineering for Success
in the Space Industry

Objectives:

- Help you …
  - understand what it takes to design, build, and test a spacecraft that works, given the unique challenges of the space industry
  - understand how developing a spacecraft within budget and schedule requires not only good engineering, but also effective teamwork and communication
  - learn important lessons from multiple case histories
  - become a better engineer!

Audience: Space-industry engineers in all disciplines and of all levels of experience

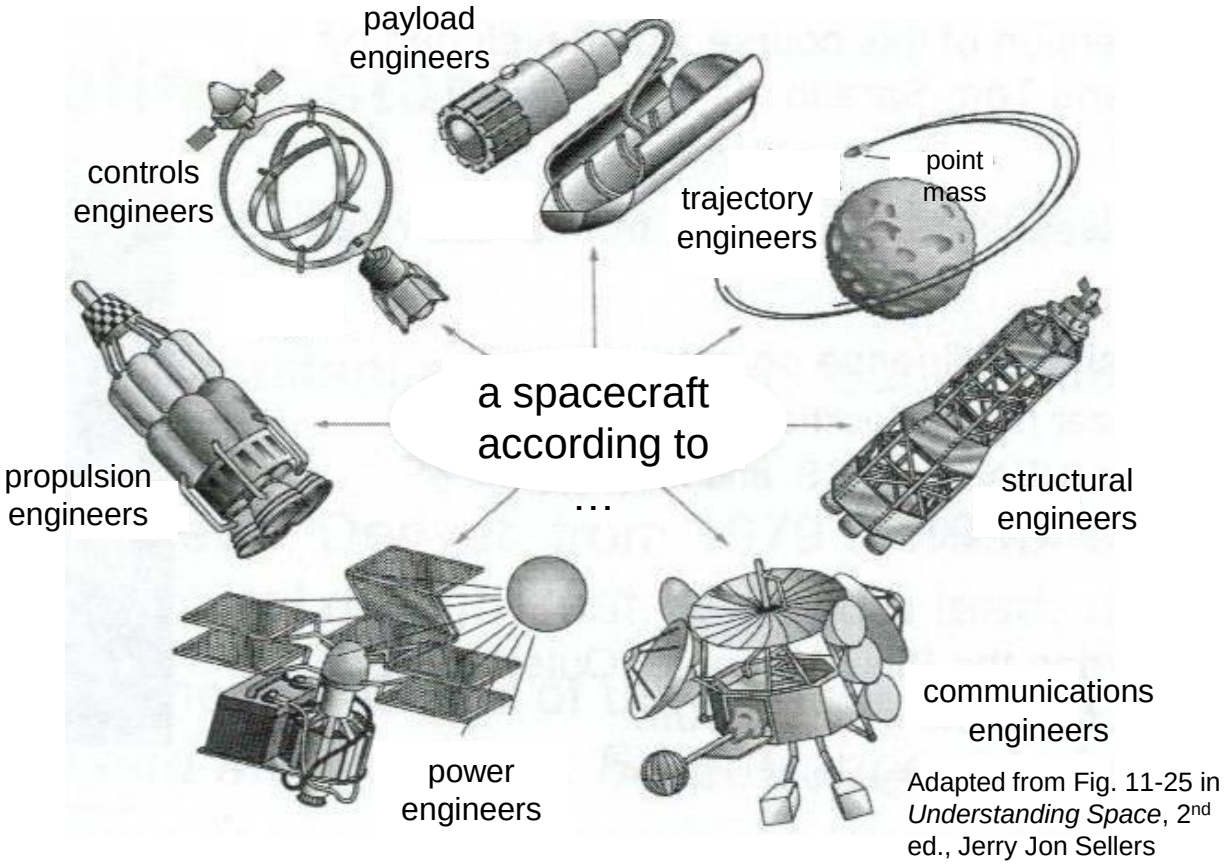
Course developed and taught by Tom Sarafin and Poti Doukas

What Constitutes a "Successful" Space Program?

In this course, a space program is considered successful if …

- the mission satisfies customer or user needs (mission success),
- program budget and schedule are met, and
- the customer-contractor-subcontractor team would be willing (and hopefully excited) to work together again.

Giving More Time and Budget to Over-specialized Engineers Does Not Work!



Adapted from Fig. 11-25 in *Understanding Space*, 2nd ed., Jerry Jon Sellers

Regardless of your area of specialty, throughout your career strive to improve your understanding of the system, the development process, and how your decisions affect cost and risk.

Adopt a "we are all systems engineers" mentality!

This course aims to help you become a more effective engineer!

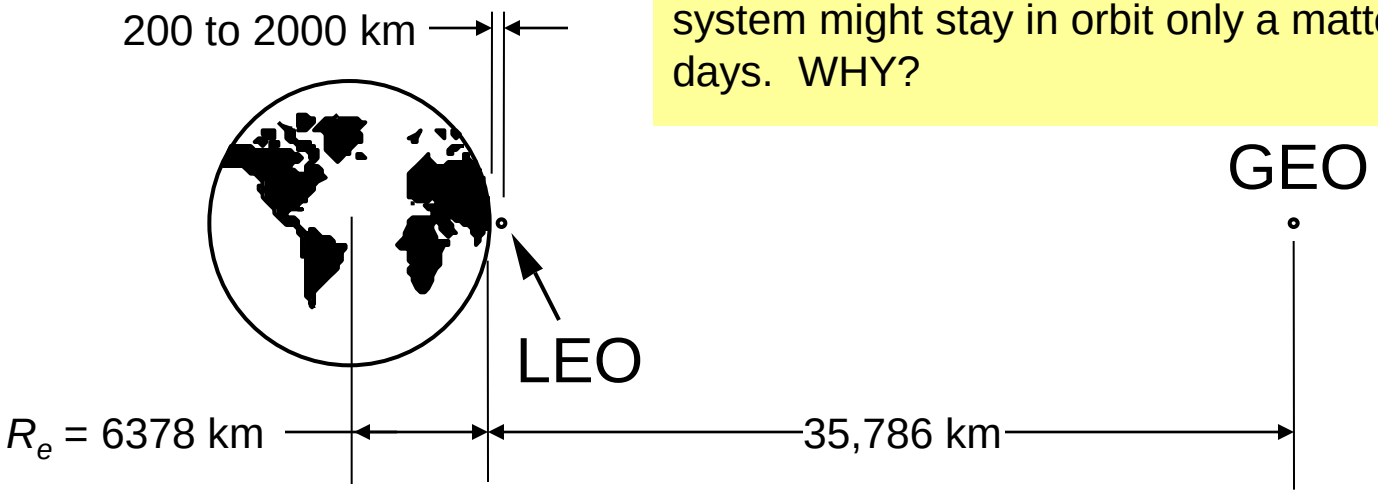
Course Topics

Introduction

1. Overview of Space Missions and Spacecraft
2. Why Are Space Missions So Challenging?
3. Nine Principles for Effective Engineering in the Space Industry
4. Understanding Requirements and Verification
5. System Development and Requirements Development
6. Reducing Cost and Risk by Design
7. Verification Planning
8. Establishing an Effective Quality System
9. Responsibly Accepting Risk
10. Communicating Effectively as an Engineer

Summary: Don’t Let the Fire Go Out!

Most Earth-orbiting Spacecraft Are at LEO or GEO



At 200 km, a satellite without a propulsion system might stay in orbit only a matter of days. WHY?

LEO = Low-Earth Orbit
GEO = Geosynchronous Orbit: satellite moves with the Earth's rotation (24-hr period)
Geostationary Orbit = equatorial GEO: satellite appears stationary over any Earth target

Where would you most likely want your satellite for these types of missions:

- **Earth observation? … LEO—close to the subject**
- **TV or radio transmission? … Geostationary—larger Earth coverage**
- **Cell phone service? … LEO—for minimal time delay**

Common Problems in Space Programs

- Problems with requirements
  - Under specified, over specified, misunderstood, frequently changing
- Problems with designs
  - Difficult to build or test, penalizing to other designs, incompatible materials or fastening hardware, changing after release, mass and power growth
- Process and people problems
  - Decisions based on wrong information, hardware not built to engineering requirements, software not ready on time, test failures that programs can't recover from, not following procedures

X-33 liquid hydrogen tank test failure, Nov. 1999

NOAA-N mishap, Sep. 2003

Common elements:

- **Problems that, in hindsight, could have been avoided**
- **Problems that seem to occur over and over again**

What Makes Space Missions So Hard?

- **If something fails during or after launch, we usually can't fix it**
  - Not like a car
  - or a cell phone
- **The space-mission customer has a huge stake in success**
  - The customer funds the system development and thus has the most to lose
  - Not like a car or a cell phone

We have to get it right the first time.

And we must convince many people before launch that we will do so.

Quality Applies to Everyone

- Think of quality as a measure of "goodness" for a product: usefulness, timeliness, dependability, value, etc.
- Everyone produces products and has customers, and everyone uses products from other people:
  - Data
  - Requirements
  - Drawings
  - Reports
  - Presentations
  - Email

Your company may have a Quality Assurance group, but ensuring quality is <u>everyone's</u> job!

- Producing products of high quality requires an understanding of your customers (product users) and how they will use those products.

A focus on quality from top down throughout an organization improves efficiency and thus reduces cost <u>and</u> increases the chances of a successful space mission.

Nine Principles for Effective Engineering in the Space Industry

1. Never stop learning, and don't become too specialized
2. Adopt the right attitude: Take responsibility for quality and mission success
3. Allow others to have ownership of their products
4. Constantly strive to improve communication and teamwork
5. Follow a sound engineering approach
6. Think ahead to avoid problems, and keep everything as simple as possible
7. Establish an effective quality system
8. Be willing to accept risks, but only those you and the other stakeholders truly understand
9. Don't let the fire go out!

What is "Verification"?

- For space programs, ***verification*** means establishing confidence before launch that the product or system will satisfy its requirements
  - "before launch"—Verification takes place while we can still change something; after launch it's usually too late (there are some exceptions)
  - "establishing confidence"—not necessarily proof
    - We can prove that our system satisfies defined, measurable criteria by analysis or test
    - But we can't prove the mission will be successful. Trying to do so would be prohibitively expensive.
  - "will satisfy its requirements"—this means we have to understand the product's requirements before attempting to verify them
    - We have to be able to distinguish between <u>product</u> requirements and <u>verification</u> "requirements"

The test is not the requirement.

Showing a positive margin of safety is not the requirement.

When applicable, they're what we do to <u>verify</u> the requirement.

Ownership and Responsibility for Procured Items

- A product (or system) ***requirement*** is something the product must do or some characteristic the product must have.
- A ***verification activity*** is something intended to demonstrate that the product or system will do what it's supposed to do.
- ***Verification criteria*** are the ground rules by which we judge the success of a verification activity.

Customer owned (or derived by contractor)

Contractor owned (but must be acceptable to customer)

Verification is part of ensuring quality, which is the contractor's job!

For a procured item, anything relating to quality or probability of success is the contractor's responsibility.

If the customer specifies how the verification must be done, the customer takes responsibility from the contractor.

If your customers do this to you, it's because they don't trust you. Strive to build their trust—your job will become much more fun!

Remember the Three Root Causes of Poor Quality

1. **Lack of understanding**
2. **Lack of care**
3. **Lack of resources**

As you progress in your career and begin to supervise or manage people, both in house and at subcontractor organizations, remember #2 in particular:

Most people naturally care when they have a sense of ownership for their products.

Don't take that ownership away by telling them how to do their jobs—which includes verification!

A Process for System Development

Step

Iterate

1. **Define objectives**
2. **Identify driving requirements**
3. **Develop concepts**
4. **Derive requirements**
5. **Evaluate concepts**
6. **Nail down requirements**
7. **Finalize design**
8. **Build system**
9. **Verify compliance**
10. **Implement system**

We start with objectives and a few driving performance requirements and constraints.

Most requirements derive from our design solution.

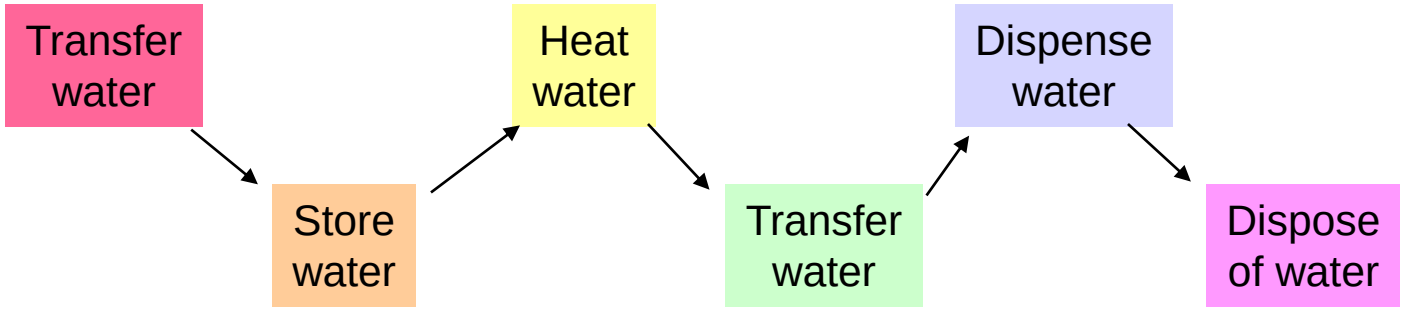
Up to a point in the process, requirements and designs should evolve simultaneously.

This process is intentionally simplified to better illustrate how requirements evolve along with the design.

Not detailed here are all the steps taken to plan verification, anticipate and prepare for downstream events such as manufacturing and handling, and establishing confidence incrementally along the way.

Requirements Stem from Functions ...

Example: Hot-water system



(functional flow block diagram, a product of functional analysis)

... and Constraints:

- cost
- schedule
- weight
- interfaces
- environments

Specification Language

- "**Shall**"—indicates a firm, binding requirement that must be verified.
  - "**Must**" has no meaning in a specification. "Must" statements will not be tracked in the requirements database.
- "**Will**"—indicates a future action, a declaration of purpose, or a statement of fact—but not a requirement.
- "**Should**" and "**may**"—mean desired or optional but not mandatory.
- Use the form "Product xxx shall (be, do, perform, or other verb)"

Phrase	Meaning
"... shall be in accordance with ..."	Must adhere to and verify
"... shall meet the intent of ..."	Not verifiable; not a good requirement
"... shall be used as a guide."	Not verifiable; not a good requirement

- Avoid indefinite or subjective terms
  - and/or, etc., maximize, suitable, adequate, best, safe

For more guidance, see Sec. 4.9 of MIL-STD-961D, Sec. 3.2 of MIL-STD-490A, and Appendix C of the NASA Systems Engineering Handbook

What's Wrong with These Requirements?

3.1.1.1 The LV first stage shall have a dry weight of 2800 lb and a gross weight of 31,500 lb.

a) Two requirements in one paragraph
b) Should be written as "not to exceed"

3.1.8 Stage One shall rely on the second-stage avionics system for attitude control.

No value added; there's a requirement implied for the avionics, but not Stage One

3.3.4 The design safety of the LV shall be 0.999.

Huh? Needs clarification, and sounds like a verification criterion rather than a requirement

3.7.2 The LV shall be assembled and integrated into a fully functioning unit.

Glad you told me!

Class Exercise

Break into multi-discipline groups, with 2 to 4 people per group.

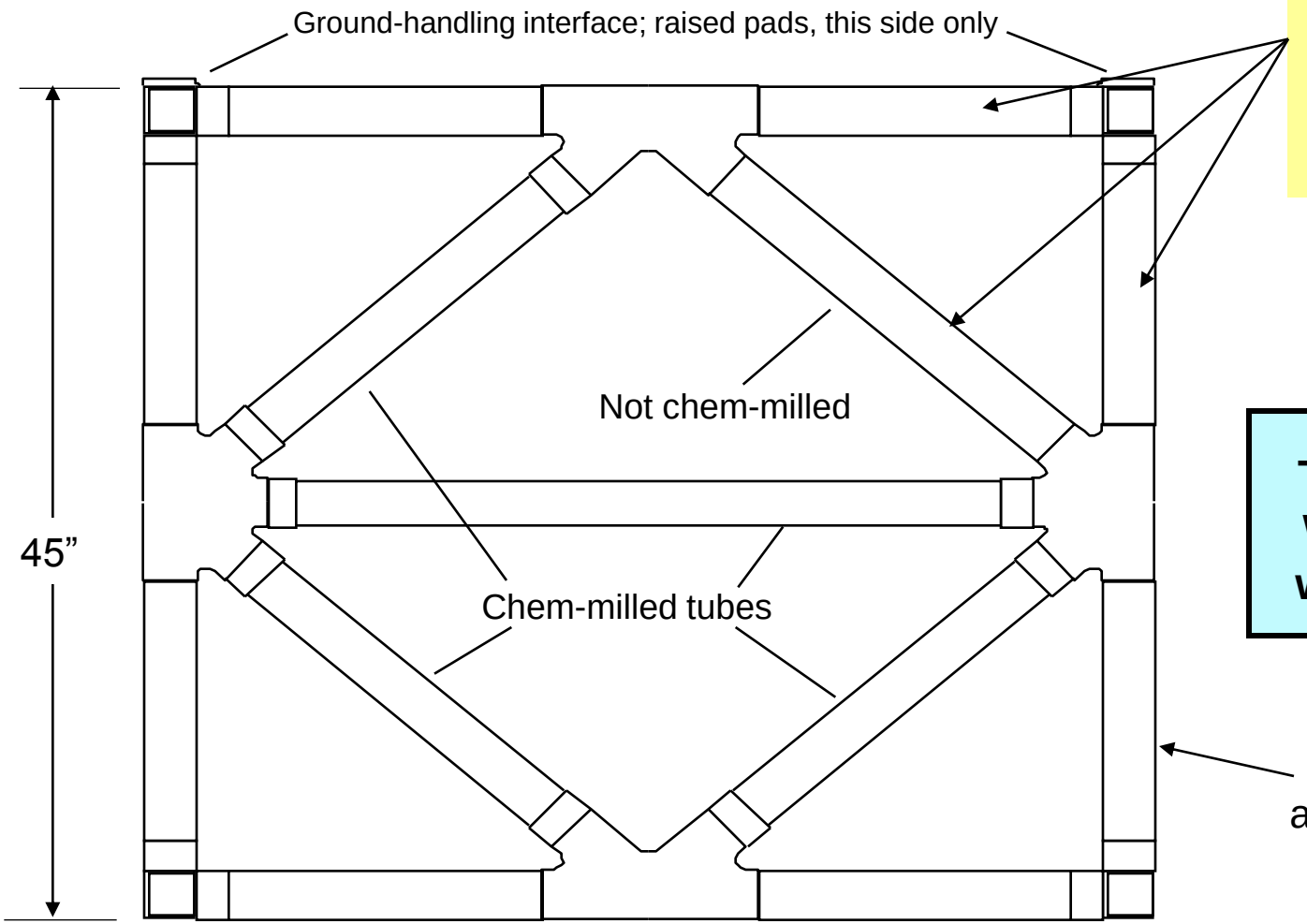
- Your group is tasked with procuring a system for ...

  (Draw from a hat)

- Pretend this item has not been invented yet
- Develop and specify the requirements for this system, making whatever assumptions you need to make
- If you're open to suggestions, try not to specify the design solution or preclude possible design solutions other than the one you are envisioning
- But if you have a clear image of exactly what you want, then fully specify the requirements necessary to ensure you get it

Example of Not Keeping it Simple

Welded frame of square tubes and machined fittings



These three members support a tripod structure for mounting a mechanism

Think ahead: What can go wrong, here?

Verification Can Be Proactive or Reactive

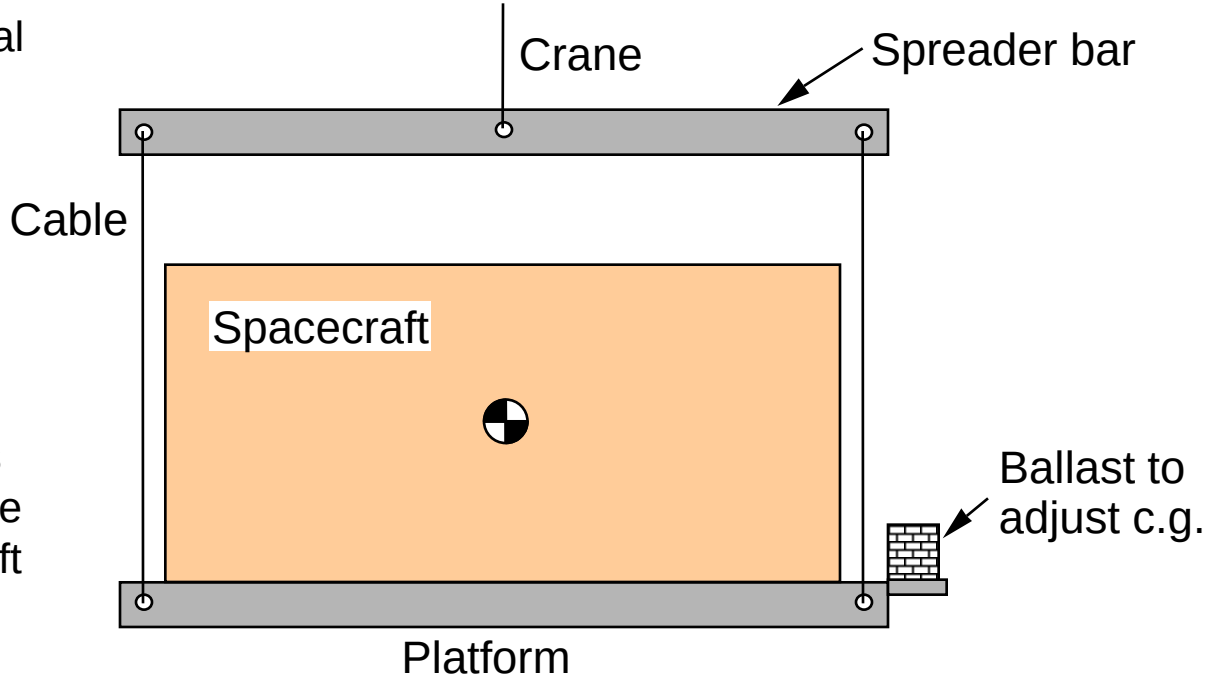
Reactive Methods (weed out poor quality)	**Proactive Methods (improve product quality)**
Analysis (when done after the design is released)	Analysis (when done prior to design release)
Inspection	Process control (goal is to learn to control a manufacturing process so well that its products do not need inspection)
End-item testing: • Qualification testing • Acceptance testing • Analysis-validation testing	Development test (to understand a problem or to improve a design or manufacturing process)

When production quantity is low, as for space programs, we can't afford to learn how to control processes as well as we'd like, so we must rely to some extent on inspections and tests.

The goal is to find the right balance.

Be Sure to Design the Right Test!

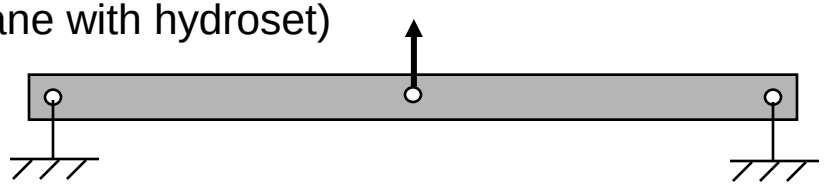
Example: mechanical ground support equipment (MGSE): lifting fixture/sling assembly



We want to structurally test this MGSE before we use it to lift our spacecraft

Test approach:

1. Proof test each cable
2. Test the spreader bar and platform by applying loads with a hydraulic jack (or a crane with hydroset)



What's wrong with this test approach?

Quality Systems at the Personal Level: Example 1

- Tom is responsible for maintaining a spacecraft-level finite element model and periodically delivering the current version to team members for trade studies or other uses and to outside organizations for coupled loads analysis (CLA) or on-orbit controls analysis.
- He recognizes the importance of maintaining control of the model's electronic file.
  - Key decisions will be based on the results of analysis with this model, so he must make sure it adequately represents the spacecraft's current design.
- He initiates a file-naming system:
  - Each time he changes the model, he gives it a new name (e.g., "SC1") and enters it into a log.
  - He documents each change in the log along with results of model checks he's made.
- Tom then alerts all potential users regarding this process
  - Makes sure they know to come to him for the most current model.
  - Instructs them to change the model name (outside of Tom's naming system) if they make a change to a delivered model, and suggests they keep a similar log.

Removing Subjectivity with Expected Cost of Failure

If we can estimate both the cost of failure and the probability of failure (a percentage), we can combine them:

Expected Cost of Failure **= Cost of Failure × Probability of Failure**

The expected cost of failure tells us how much we should be willing to spend to eliminate a risk.

Estimating probability of failure requires a meaningful statistical database, as is used for reliability analysis

Hypothetical Problem: Responding to a Negative Structural Margin of Safety

- A spacecraft is about to complete integration and test, two months before launch.
- The verification loads cycle shows increased loads, causing a negative 20% margin of safety (–0.20) for ultimate failure of a key member in the primary structure.
  - If the member fails, the mission will be lost. (Human safety is not at issue.)
- All unnecessary conservatism has been scrubbed from the stress analysis and the loads analysis; the analysis is relatively straightforward and the failure mode is well understood.
- The ultimate factor of safety used in the analysis is 1.25. Limit load is estimated at 3-sigma probability (99.87%), and the allowable stress is A-basis (99%).
- The structure was tested, but to 1.1 times the original limit loads, which were 35% lower than the new limit load.
- This is a one-of-a-kind structure; there is no qualification unit we can test.
- Redesign is out of the question—replacing the structure with a redesigned one would cause the program to miss its launch window.
- Structural reinforcement could be added. Total estimated cost: $200,000 (labor, materials, overhead)
- Mission value = $100,000,000

What would you do?

The Quality of a Report or a Presentation Reflects the Quality of the Engineering

If your report or presentation ...	then your audience infers ...	which means ...
is disorganized,	you may have missed something,	they don't trust your conclusions.
doesn't clearly define the problem,	you don't understand the problem,	they don't trust your conclusions.
doesn't state any applicable requirements and acceptance criteria,	you didn't understand the requirements and criteria, and thus you didn't satisfy them,	they don't trust your conclusions.
doesn't define the solution process,	the process you used is irrational,	they don't trust your conclusions.
doesn't explain your assumptions and the steps you took to confirm them or assess sensitivity,	the assumptions you made are irrational or incorrect,	they don't trust your conclusions.
doesn't refer to sources of key data,	the data is outdated or invalid,	they don't trust your conclusions.
doesn't document each step,	you made a mistake,	they don't trust your conclusions.
doesn't describe the checks you made,	you didn't find the mistake,	they don't trust your conclusions.