

Engineering Disasters

- **How projects fail, why they fail and how to prevent failure.**
- **All in one easy lecture!**
 - **DuRant Lewis, Atmel Corporation**

Engineering Disasters

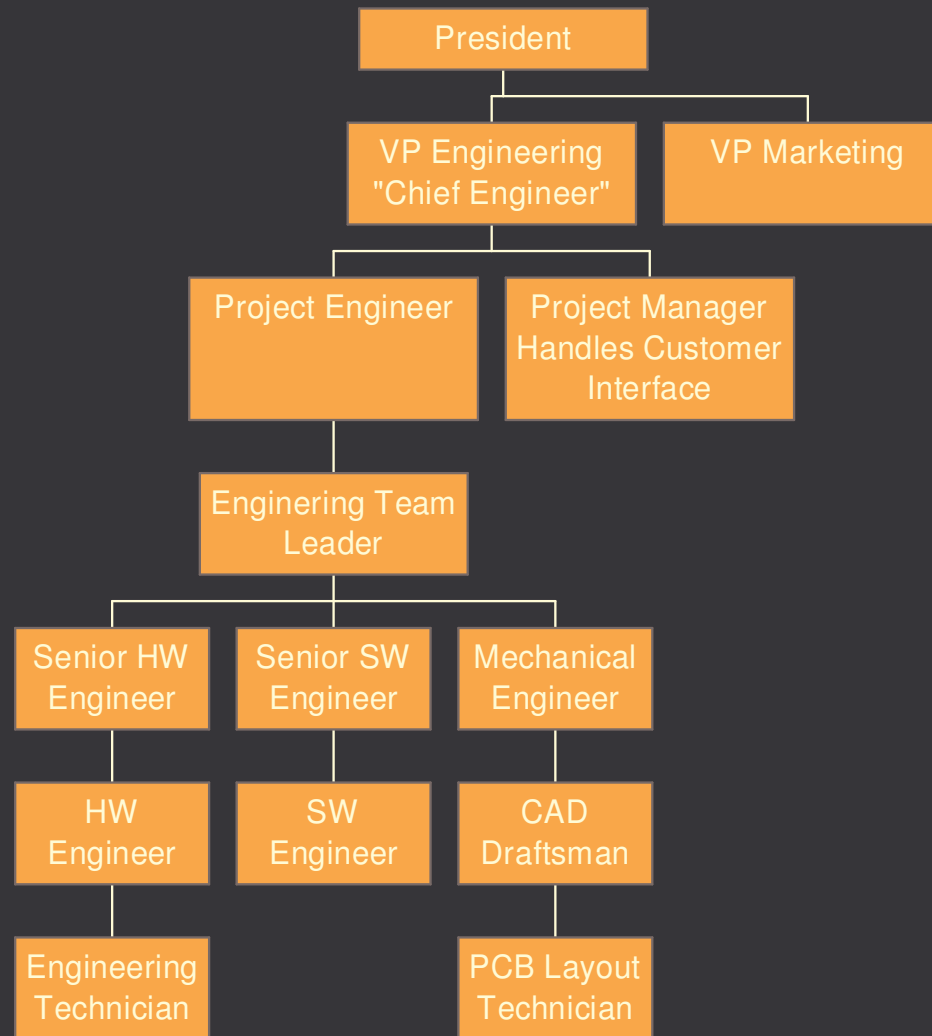
- A Night at the Whitney!



Engineering Disasters

- To Understand why engineering projects fail, one must first take a look at the structure of the typical engineering department. So let's take a look:

Typical Engineering Department



Other Important Parts

Component Engineering
Part numbers/SCD's
End-of-Life Issues

Field Services

Compliance Engineering
FDA/FCC/FAA/UL/

Production Engineering
Transition to Production

An aerial photograph of a coastal region. In the foreground, there is a large body of water with some small, rocky islands. In the middle ground, a larger island with some buildings and a pier is visible. In the background, a large, rugged mountain range stretches across the horizon under a clear sky.

Why projects fail

- Projects fail for four principal reasons:
 - Cost
 - performance
 - schedule
 - catastrophic failure

What can go wrong - The Endless Revision Cycle

- A design is flawed, so it's re-spun. Some original problems are fixed, but the changes cause new problems. So it's re-spun, and the same thing happens.
- The product never seems to get completely fixed.

What can go wrong - The Endless Revision Cycle

- This cycle is hard to break - so it's best to prevent it.
 - Simulate circuit behavior - spot problems before the design is fabricated.
 - Control revisions, keep a revision history and have a release process.
 - Do design reviews. Peer review helps.

What can go wrong - The Endless Revision Cycle

- Piece-wise test as much of the design as possible, with both simulation and HW emulation.
- Use a portion of existing designs known to work - don't re-invent everything on one project.
- Do design reviews. Again, peer review helps.

What can go wrong - Project is doomed at the start

- Many projects are doomed before they start due to poor planning, improper resources, or a poor specification.
 - If key assets are missing, like a senior engineer, the project might not be spec'ed correctly.
 - Proper design tools and test equipment are missing
 - Good tools save engineering time, and are worth the cost.

What can go wrong - Project is doomed at the start

- Tools - Continued
 - The cost of “free” tools is the time it takes to get it to work.
 - Support is important In the case of a commercial tools vendor, they offer software updates. In the case of an open source tool, it’s a posting on a bulletin board.
 - A full set of tools [IDE, compiler, RTOS, TCP/IP stack for an ARM design costs 30k. But how fast can you burn this in engineering time?

What can go wrong - Project is doomed at the start

- **Tools - Continued part II**
 - **If the company does not purchase the proper simulation tools (thermal, electrical, structural), then**
 - **Engineers guess when they design, and they tend to guess conservatively, increasing cost.**
 - **The system might fail because a design flaw is not detected, then more spins as they design by trial and error.**

What can go wrong - Project is doomed at the start

- Poor specifications can doom a project
 - If you work with other companies, they must work to a clear performance specification
 - The Rockwell B1B example.
 - Be clear about the specifications going in - Don't introduce late-breaking features into a product late in the development cycle. The cost will skyrocket, and performance could suffer.
 - Not enough testing.

What can go wrong - Improper use of consultants

- There is a time to use consultants, but keep the “Tribal Knowledge” in house.
 - There is a time to use them - work load issues or areas where a company does not have any expertise.
 - In an ideal situation, do the first couple of projects with a consultant, then transfer it it-house.
 - Don’t use consultants to the point that the company cannot exist without them - ongoing support and product updates can be a problem.

What can go wrong
Great design, but can't be produced

- Always design for production. The best design ever is no good if it can't be produced
 - Involve Manufacturing in the design and prototype process
 - Manufacturing will tell you what's wrong, but you have to ask.
 - This is more important now with remote manufacturing.

What can go wrong

Great design, but can't be produced

- Make the assembly as easy as possible
 - Streamline jumpering and configuration
 - SCSI panel example.
 - Allow relief's around mounting holes to clear tools
 - Hide or shield battery nodes.
 - Use self-calibration techniques.
- Work with compliance engineers to help with FDA, FCC and UL
 - Source suppression of EMI is much cheaper than shielding after the fact.
 - UL power supply compliance is best built in.

What can go wrong

Great design, but can't be produced

– Examples:

- GM-10 vs. Ford Taurus (1989)
 - 44 hours assembly vs. 24 hours. GM lost \$900.00 on each car.
- Airflow sensor example
 - Can't be calibrated.
- German Tiger vs. American Sherman.
 - 100,000 hours to build a Tiger, it's no wonder Germany produced a little over 1,200 Tiger II tanks
 - America make about 90,000 Shermans.

Engineering Disasters

What you can do:

- At the worker bee level, there are many factors that are out of your control. But some you can affect.
 - Know the Schedule.
 - Engineers who are aware of the deadlines will work to meet them
 - Have a backup plan in case something does not work out. - Alternative vendor, consultant, etc

Engineering Disasters

What you can do:

- **Design within the specifications.**
 - Many designs just happen to work, like FPGA designs that are not constrained, or analog designs that use features in excess of the data sheet.
- **Use concurrent engineering.**
 - For example, spec the layers of a PCB and design the firmware while the board is out for fab.
- **Use programmable logic or micro's**
 - It's an easy place to fix problems or revise the board

Analyzer

Waveform Machine 1

Acq. Control

Print

Run

Accumulate
Off

Current Sample Period = 8.000 ns
Next Sample Period = 8.000 ns

sec/Div
333 ns

Delay
1.464 ns

Markers
Off

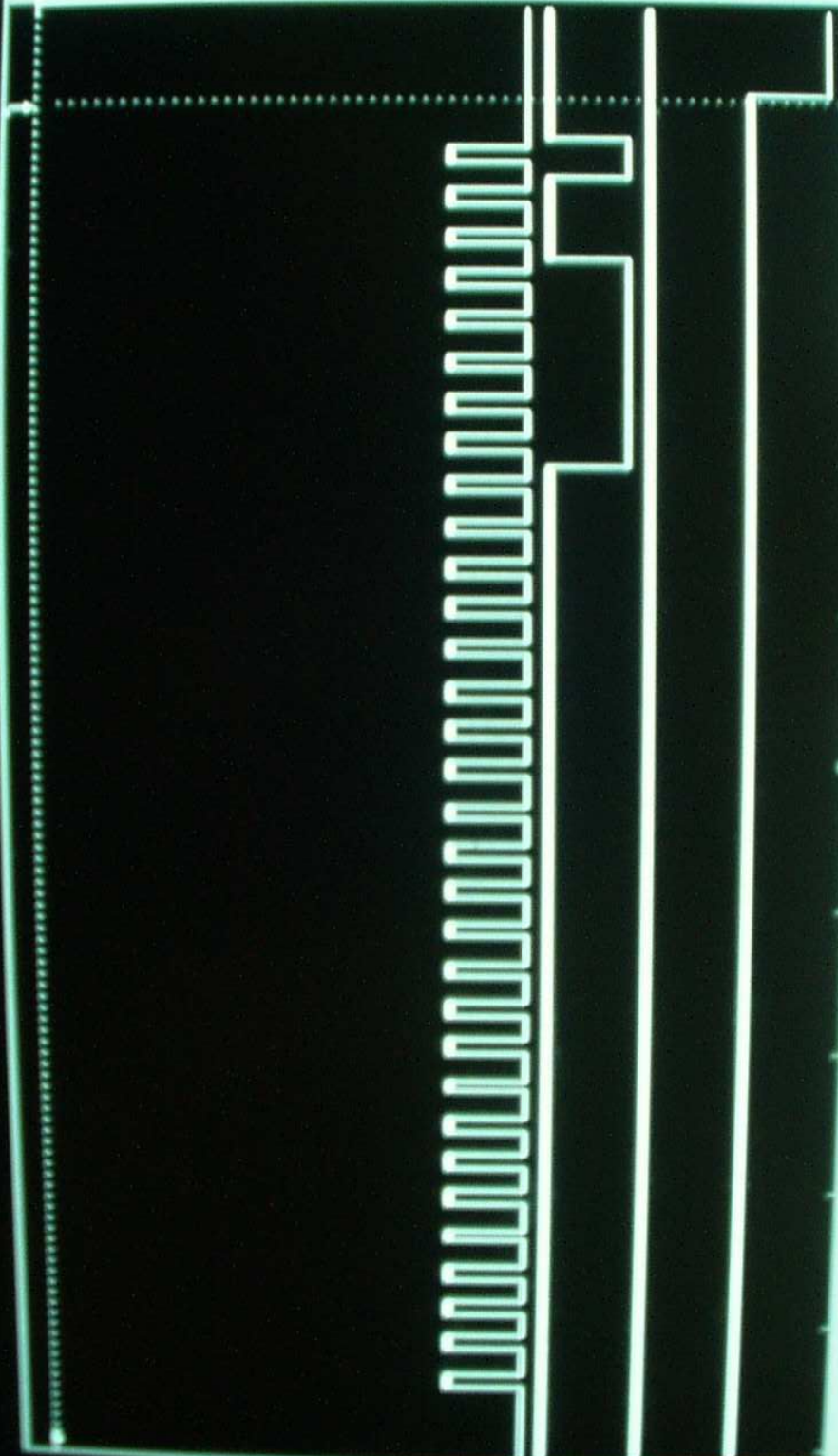
Acquisition Time
04 Jun 2005 15:58:33

CS_N

MISO

MOSI

SCLK



Engineering Disasters

What you can do:

- Document, Document, Document.
 - Keep revision histories and communicate with your peers. Don't code in the dark recesses of the syntax for a programming language. If it's easy for a human to read, compilers will be efficient also.
 - ISO 9000 requires it.
- Know that worker bees will tend to underestimate the time to complete a task by a factor of 4, so give yourself the time to do a job.

Engineering Disasters

What you can do:

- Watch out for bad companies.
 - Success is a habit and so is failure. There are miserable companies that never really succeed but never really go broke either. One good way to avoid such companies is to talk to vendors. If a company owes vendors - that's a bad sign. Many company records are on the web.
- Look out for Engineering management by non-engineers (SPECIAL HANDLING REQ'D)
 - Many non-engineers do not understand the debug process, or why time frames are hard to predict.