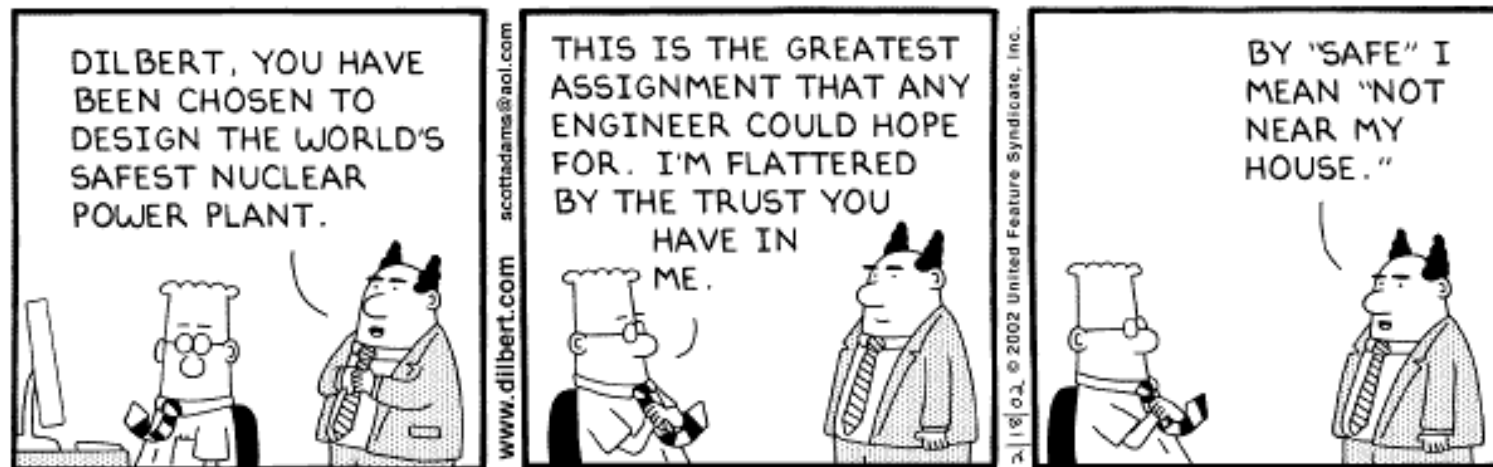


# Computer Risks, Safety, and Designing for Humans



Copyright © 2002 United Feature Syndicate, Inc.

# Today's Agenda

- consider origins of human factors design
- frighten you with some gory stories about what happens when human factors are ignored
- discuss importance of designing for human users

# Origins of Human Factors

- WW II: invention of machines that taxed human sensorimotor abilities
- even after extensive training, frequent (fatal) errors

⇒ example:

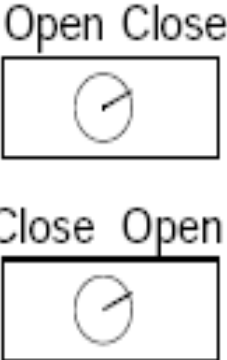
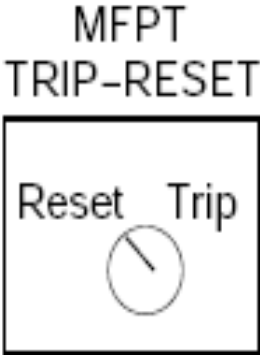
if airplane booster pump fails, turn on fuel within 3 seconds

# Problem

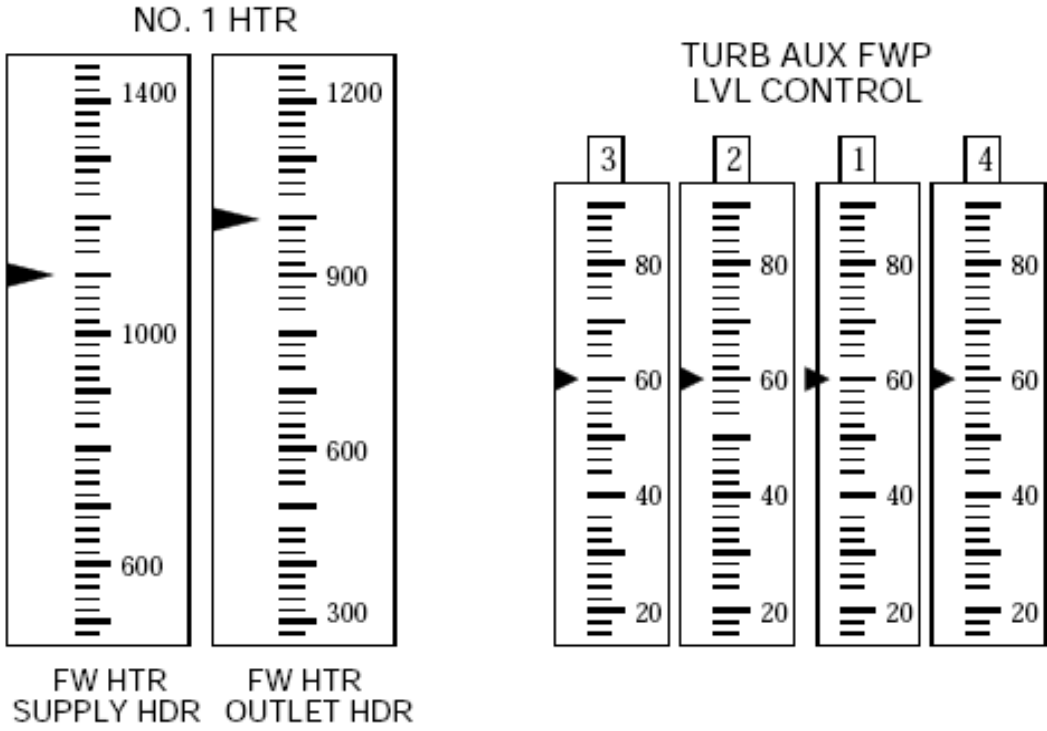
**It actually took  
pilots 5 seconds!**



# Actual Switches in Power Plant

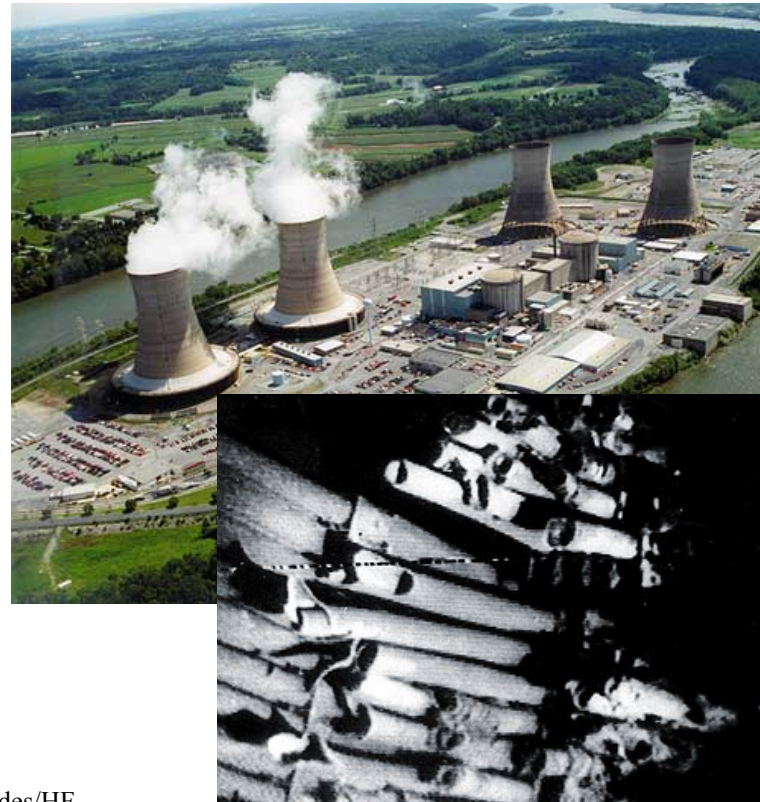


# Any problems here?



# Three Mile Island (1979)

- Some sensors were not included in the design of the plant to save engineering time and money



Credit: Michael Kalsher - [http://www.kalsher.com/hf/slides/HF%202008\\_PP\\_Ch01.ppt](http://www.kalsher.com/hf/slides/HF%202008_PP_Ch01.ppt)

The reactor core - 4 years later

# China Airlines Flight 006 (1985)

“... a China Airlines 747 suffered a slow loss of power from its outer right engine. This would have caused the plane to yaw to the right, but the autopilot compensated, until it finally reached the limit of its compensatory abilities and could no longer keep the plane stable...”

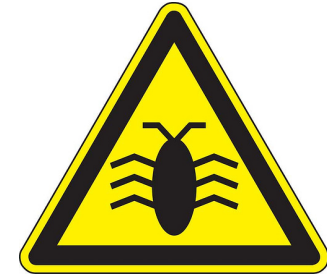


# Therac-25 (1985-1987)

- software wasn't blamed at first:



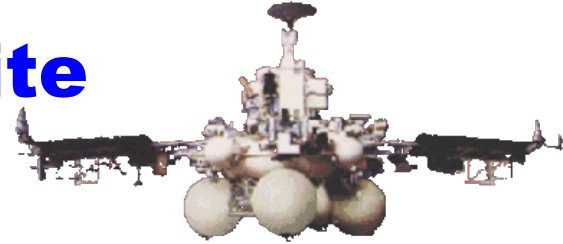
# Complacency



*"Programming errors have been reduced by extensive testing on a hardware simulator and under field conditions on teletherapy units. Any residual software errors are not included in the analysis... Program software does not degrade due to wear, fatigue, or reproduction process."*

# USSR Phobos I Satellite

## Launched July 7, 1988



Norman says:

"... not long after launch, a ground controller omitted a single letter in a series of digital commands sent to the spacecraft. And *by malignant bad luck*, that omission caused the code to be mistranslated in such a way as to trigger the [ROM] test sequence [that was intended to be used only during checkout of the spacecraft on ground]"

## Don Norman's take



*“There are two things we know about unexpected conditions:*

# Normal accidents



High-risk systems:

Normal (“inevitable”) accidents:

# Risks of Automation

- doesn't remove people from systems; it moves them to different responsibilities but makes decision-making more difficult
- system response to problems can hide underlying cause, e.g., autopilot in China Airlines 006

# Feedback Limitations

- can't convey entire internal state to user
- limited quality of feedback

# **Mechanical vs. Software systems**

- mechanical failure
  - techniques to reduce, detect, mitigate
- software failure
  - abstract, typically based on design problems



# Context-sensitivity

- ATC: US Software used in UK flights crossing 0° longitude
- F-16 navigation control used over Dead Sea

# Designing for Error

- understand the causes of error
- design to minimize these causes
- make it possible to \_\_\_\_\_
- make it hard to perform \_\_\_\_\_
- make it easy to discover errors
- change attitude toward errors

**Don't:**

# **Dangers of “human-in-loop”**

- software errors mix-up patient data/names
- incorrect output of lab diagnostic devices

## **Murder by Computer Error (1988)**



“A print-out error caused a medical insurer to convince a 54-year old woman that she suffered from a fatal form of syphilis, and that she had transmitted the disease to her children; in panic, she strangled her 15-year old daughter and tried to kill her 13-year old son. The boy escaped, and succeeded in preventing his mother's death from a drug overdose. The Dusseldorf court dismissed the accusation of murder, laid all blame with the computer error, and declared the woman unaccountable for her actions.”

## **Dangers of “computer-in-loop”**

- computers add difficulty to accident investigations
- “malfunctioning electronics will not be found in wreckage”