Richard W. Hamming

Learning to Learn

The Art of Doing Science and Engineering

Session 27: Systems Engineering

The Big Picture

Parables:

- "I'm helping to build a cathedral."
- "I'm educating students for their future."

Most of the time a person is so immersed in the details, they fail to see the big picture.

- Essence of bureaucracy

Systems engineering is the attempt to look at the whole as a whole at all times.

Challenge of System Engineering

Knowing the big picture without knowing how to work the details properly

- Example: school students learn & prepare for future, but still cram for finals which is counterproductive

Translate local actions in global actions

Also applies to one's career and life goals

- Expanding circle of obligations (team, department, company, career, etc.) based on time and importance

Rule #1 of System Engineering

If you optimize individual components, you will probably ruin the system performance

- Hamming's differential analyzer experience
- Students cramming for finals
  - *Could they pass a capstone test on their major/minor fields?*
- College mathematics divided into discrete topics
  - *Loses the interconnectivity of mathematical ideas*

Successes in Systems Engineering

Examples:

- Streamlined, just-in-time, Venetian shipbuilding
- Telephone company – designed & operated system
- Economies of scale

Key is to build flexibility into the design

- This is true about software engineering as well

Rule #2 of Systems Engineering

Part of systems engineering design is to prepare for changes, so that they can be gracefully made and still not degrade the other parts

- Example: learn fundamentals to avoid obsolescence
- Will help when field changes are made to the system

Rule #3 of Systems Engineering

The closer you hit design specifications, the worse performance will be when overloaded.

- An “optimized” 30 Ton bridge will fail at 1 pound over
- Gracefully degradation is essential when design specifications are exceeded

Systems Engineering Teams

Specialists brought together to form a team is the basis of systems engineering.

Specialists must be able to go back to their technical areas to maintain skills.

System Engineering is Never Done

Individual’s idea of what ought to be done can run counter to system optimization.

- Example: computer users cheat system by dividing up large programs into small ones to meet time limits
- Example: changing rules for promotion changes your behavior, despite what may be best for the company

- **Deadlines limit optimization of the design**
- Role of the engineer

Clients and Acceptance

Client knows symptoms of the problem, but not the cause

- Job of the system engineer is to find the root cause
- Customer will place demands on system engineer

Boundaries of the solution are elastic

- Accept there is never a definite solution to a problem
- Can’t be formally taught, experienced in a laboratory
- What works in the lab can’t be done in the field

Observations

Hamming has often seen the wrong problem solved exactly right

- Doesn’t help the customer
- Better to have a poor solution to the right problem

The problem always changes

- Example: Nike test kills a single straight & level drone at known altitude. But, the actual problem was coordination of shooting down multiple planes by multiple missiles to protect highly valued targets.

More Observations

Each solution iteration must bring about a deeper understanding of the problem.

The trouble with systems engineering is the human being must be taken into account.

- Example: Service versus cheap service: Automated telephone system loses usefulness over human operator
- Individuals will optimize their performance to the detriment of the optimization of the system