

## Complexity Revisited

6.033 Lecture 25

May 8, 2000

Lecturer: Jerry Saltzer

Saltzer@mit.edu

<http://mit.edu/Saltzer>

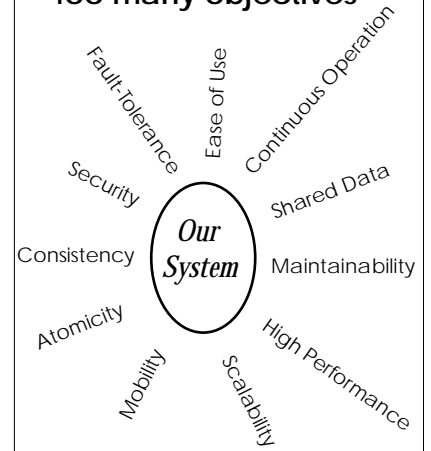
Saltzer, 5/8/2000, slide 1

## Coping with Complexity

- Sources
- Learning from disaster (and experience)
- Fighting back
- Admonition

Saltzer, 5/8/2000, slide 2

## Too many objectives



## Not enough principles

Saltzer, 5/8/2000, slide 3

Many objectives

+

Few principles

+

High  $d(\text{technology})/dt$

=

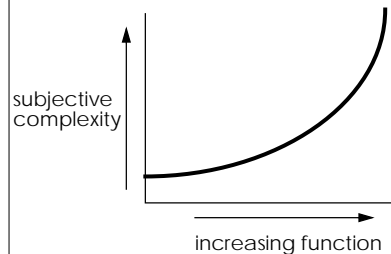
Very high risk

*The Tar Pit*

Saltzer, 5/8/2000, slide 4

## No Hard-edged barrier—

it just gets worse...



Saltzer, 5/8/2000, slide 5

## Learn from failure

(photo)

Pyramid at Meidum

(photo)

Bent Pyramid

Saltzer, 5/8/2000, slide 6

## Learn from failure

*Complex systems fail for  
complex reasons*

Find the cause

Find a second cause

Keep looking

Find the mind-set

(see Petroski, *Design Paradigms*)

Saltzer, 5/8/2000, slide 7

## NYC control of 10,000 traffic lights

Univac, based on experience in Baltimore and Toronto

started: late 1960's  
scrapped: 2-3 years later  
spent: ?

- second-system effect:
  - new radio control system
  - new software
  - new algorithms
- based on systems 100X smaller, incommensurate scaling

Saltzer, 5/8/2000, slide 8

## California Department of Motor Vehicles

Vehicle Registration,  
Driver's License

started: 1987  
scrapped: 1994  
spent: \$44M

- underestimated cost by factor of 3
- slower than 1965 system
- governor fired the whistleblower
- DMV blames Tandem
- Tandem blames DMV

Saltzer, 5/8/2000, slide 9

### United Airlines/Univac

automated reservations,  
ticketing, flight  
scheduling, fuel delivery,  
kitchens, and general  
administration

started: late 1960's  
scrapped: early 1970's  
spent: \$50M

- second system: tried to automate everything, including the kitchen sink

(ditto: Burroughs/TWA)

Saltzer, 5/8/2000, slide 10

### CONFIRM

Hilton, Marriott, Budget,  
American Airlines

Hotel reservations with  
links to Wizard and Sabre

started: 1988  
scrapped: 1992  
spent: \$125M

- Second system
- Very dull tools (machine language)
- Bad-news diode
- See CACM October 1994, for details

Saltzer, 5/8/2000, slide 11

### Advanced Logistics System

U.S. Air Force  
Materiel and transport  
tracking

started: 1968  
scrapped: 1975  
spent: \$250M

- second system effect

Saltzer, 5/8/2000, slide 12

### SACSS(California) State-wide Automated Child Support System

Started: 1991 (\$99M)  
"on hold": Sept. 1997  
Cost: \$300M

- "Lockheed and HWDC disagree on what the system contains and which part of it isn't working."
- "Departments should not deploy a system to additional users if it is not working."
- "...should be broken into smaller, more easily managed projects..."

Saltzer, 5/8/2000, slide 13

### Taurus

British Stock Exchange

Share trading system

started: ?  
scrapped: 1993  
spent: £400M = \$600M

- "massive complexity of the back-end settlement systems..."
- delays and cost overruns

Saltzer, 5/8/2000, slide 14

### IBM Workplace OS for PPC

Mach 3.0 + binary  
compatibility with Pink,  
AIX, DOS, OS/400 + new  
clock mgt + new RPC +  
new I/O + new CPU

Started: 1991  
Scrapped: 1996  
Spent: \$2B

- 400 staff on kernel, 1500 elsewhere
- "sheer complexity of the class structure proved to be overwhelming"
- big-endian/little-endian not solved
- inflexibility of frozen class structure

Saltzer, 5/8/2000, slide 15

### Tax Systems Modernization

U.S. Internal Revenue  
Service, replaces 27  
aging systems

Started: 1989 (est.: \$7B)  
Scrapped: 1997?  
Spent: \$4B

- all-or-nothing massive upgrade
- government procurement regulations

Saltzer, 5/8/2000, slide 16

### Advanced Automation System

U.S. Federal Aviation  
Administration

replaces 1972 Air Route  
Traffic Control System

started: 1982  
scrapped: 1994  
spent: \$6B

- changing specifications
- grandiose expectations
- congressional meddling

Saltzer, 5/8/2000, slide 17

### London Ambulance Service

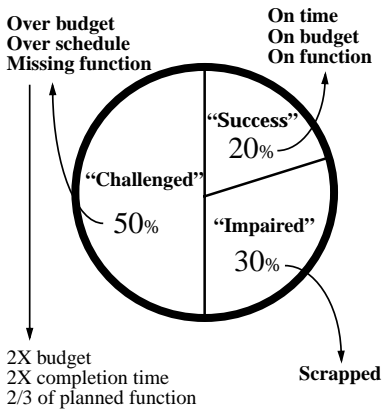
Ambulance dispatching

started: 1991  
scrapped: 1992  
cost: 20 lives lost in 2 days  
of operation, \$2.5M

- unrealistic schedule (5 months)
- overambitious objectives
- unidentifiable project manager
- low bidder had no experience
- backup system not checked out
- no testing/overlap with old system
- users not consulted during design

Saltzer, 5/8/2000, slide 18

### 1995 Standish Group Study



Saltzer, 5/8/2000, slide 19

### Recurring problems

- Incommensurate scaling
- Too many ideas
- Mythical man-month
- bad ideas included
- modularity is hard
- bad-news diode

Saltzer, 5/8/2000, slide 20

### Why aren't abstraction, modularity, hierarchy, and layers enough?

- First, you must understand what you are doing.
  - It is easy to create abstractions; it is hard to discover the *right* abstraction.
- (ditto for modularity, hierarchy, and layers)

Saltzer, 5/8/2000, slide 21

### Fighting Back: Control Novelty

Sources of excessive novelty...

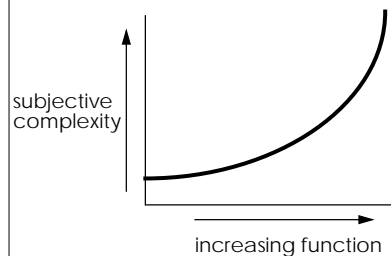
- second-system effect
- technology is better
- idea worked in isolation
- marketing pressure

Some novelty is necessary; the hard part is figuring out when to say **No**.

Saltzer, 5/8/2000, slide 22

### No Hard-edged barrier—

it just gets worse...



Saltzer, 5/8/2000, slide 23

### Fighting Back: Control Novelty

- Something simple working soon
- One new problem at a time

Saltzer, 5/8/2000, slide 24

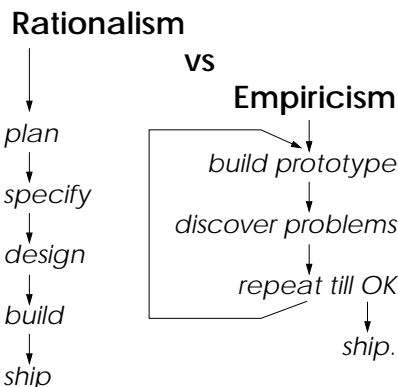
### Fighting Back: Feedback

Design for Iteration,  
Iterate the Design

- Something simple working soon
- One new problem at a time
- Find ways to find flaws early
- Use iteration-friendly design
- Bypass the bad-news diode
- General: Learn from failure

Saltzer, 5/8/2000, slide 25

### Brooks's version:



(stolen from Brooks, 1993)

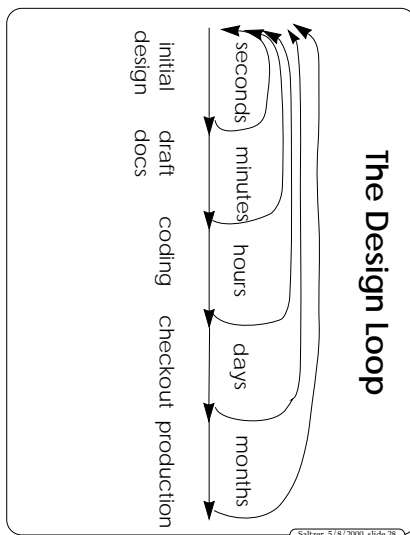
Saltzer, 5/8/2000, slide 26

### Fighting Back: Find bad ideas fast

- Understand the design loop
- Examine the requirements  
“and ferry itself across the Atlantic”  
(LHX light attack helicopter)
- Try ideas out—but don't hesitate to scrap them

*Requires strong, knowledgeable management*

Saltzer, 5/8/2000, slide 27



### Fighting Back: Find flaws fast

- Plan, plan, plan
- Simulate, simulate, simulate
- design reviews, coding reviews, regression tests, performance measurements
- design the feedback system  
e.g., alpha test, beta test, no-penalty reports, incentives & reinforcement

Saltzer, 5/8/2000, slide 29

### Use Iteration-friendly design methods

- Authentication logic (Ch 6)
- Alibis (space shuttle paper)
- Failure tolerance models (Ch 7)

General method:

- document all assumptions
- provide feedback paths
- when feedback arrives, review assumptions

Saltzer, 5/8/2000, slide 30

### Fighting Back: Conceptual integrity

- One mind controls the design
  - *Reims cathedral*
  - *Macintosh*
  - *Visicalc*
  - *SunOS*
  - *X Window System*
- Good esthetics yields more successful systems
  - *Parsimony*
  - *Orthogonality*
  - *Elegance*

Saltzer, 5/8/2000, slide 31

### Obstacles

- Hard to find the right modularity
- Tension: need the best designers—but they are the hardest to manage
- The Mythical Man-Month

Saltzer, 5/8/2000, slide 32

### Fighting Back: Summary

- Control novelty
- Install Feedback
- Find bad ideas fast
- Use iteration-friendly design methods
- Conceptual integrity

Saltzer, 5/8/2000, slide 33

### Admonition

Make sure that none of the systems *you* design can be used as disaster examples in future versions of this talk.

Saltzer, 5/8/2000, slide 34

### 6.033 Theme song

'Tis the gift to be simple, 'tis the gift to be free,  
'Tis the gift to come down where we ought to be;  
And when we find ourselves in the place just right,  
'Twill be in the valley of love and delight.

When true simplicity is gained  
To bow and to bend we shan't be ashamed;  
To turn, turn will be our delight,  
Till by turning, turning we come out right.

— *Simple Gifts*, traditional Shaker hymn

Saltzer, 5/8/2000, slide 35