SYSTEM ENGINEERING AND SOFTWARE EXCEPTION HANDLING

Herb Hecht
SoHaR Incorporated
Culver City, California

WHY WE ARE HERE

1. Many failures in critical systems are due to missing or faulty exception handling
   and we want to change that
2. They were not tested under the exception conditions
3. The requirements were not specific about exceptions that had to be tolerated
4. Comprehensive specification of exceptions that have to be tolerated is difficult – or is it impossible?

HOW SOFTWARE FAILS

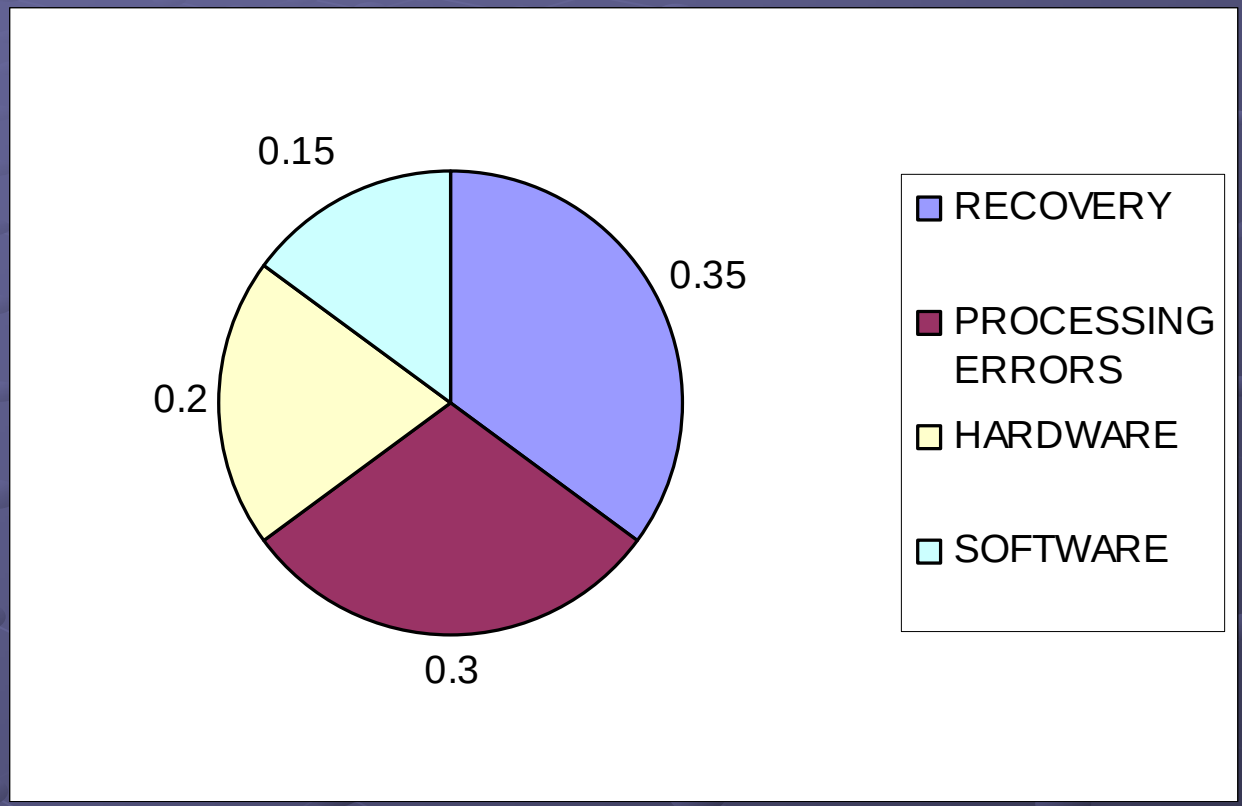
"The main line software code usually does its job. Breakdowns typically occur when the software exception code does not properly handle abnormal input or environmental conditions – or when an interface does not respond in the anticipated or desired manner."

C. K. Hansen, *The Status of Reliability Engineering Technology 2001*, Newsletter of the IEEE Reliability Society, January 2001

SOME SPECTACULARS

- THERAC-25 FATAL RADIATION OVERDOSES
  - DID NOT SUPPRESS OPERATOR INPUT WHILE MAGNETS WERE REPOSITIONED
- ARIANE 5 CRASHED AFTER LAUNCH
  - DISABLED LANGUAGE PROVIDED EXC. HANDL.
  - PERMITTED SHUT-DOWN OF BOTH NAV SYST.
- MARS POLAR LANDER HARD LANDED
  - FAILURE TO DE-BOUNCE CONTACTS

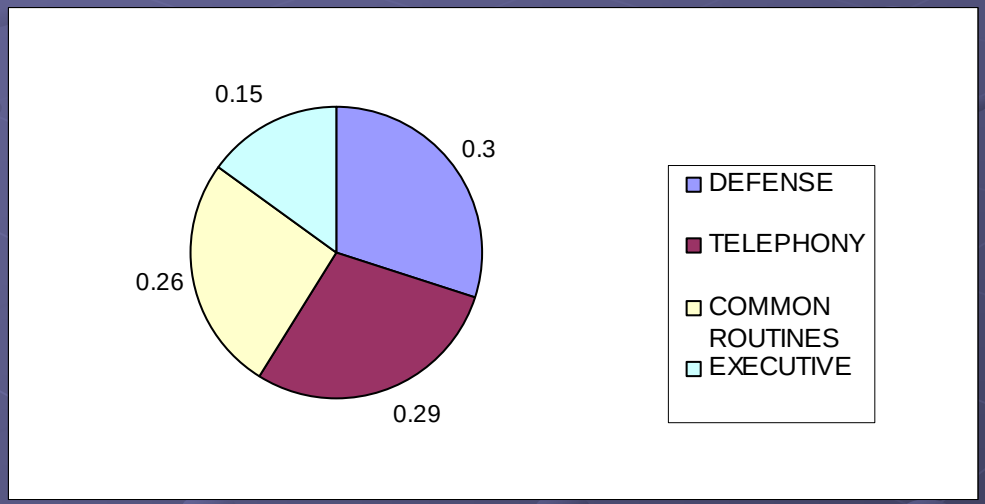
IMPORTANCE OF EXCEPTION HANDLING - 1



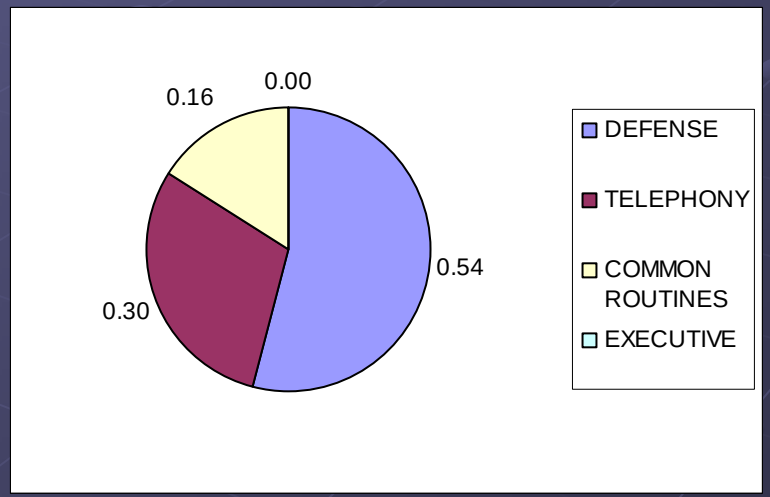
Toy, W. N., “Fault-Tolerant Design of AT&T Telephone Switching Systems” in *Reliable Computer Systems: design and evaluation,* Siewiorek and Swarz, eds., Digital Press, Burlington MA, 1992

IMPORTANCE OF EXCEPTION HANDLING - 2

ALL FAILURES

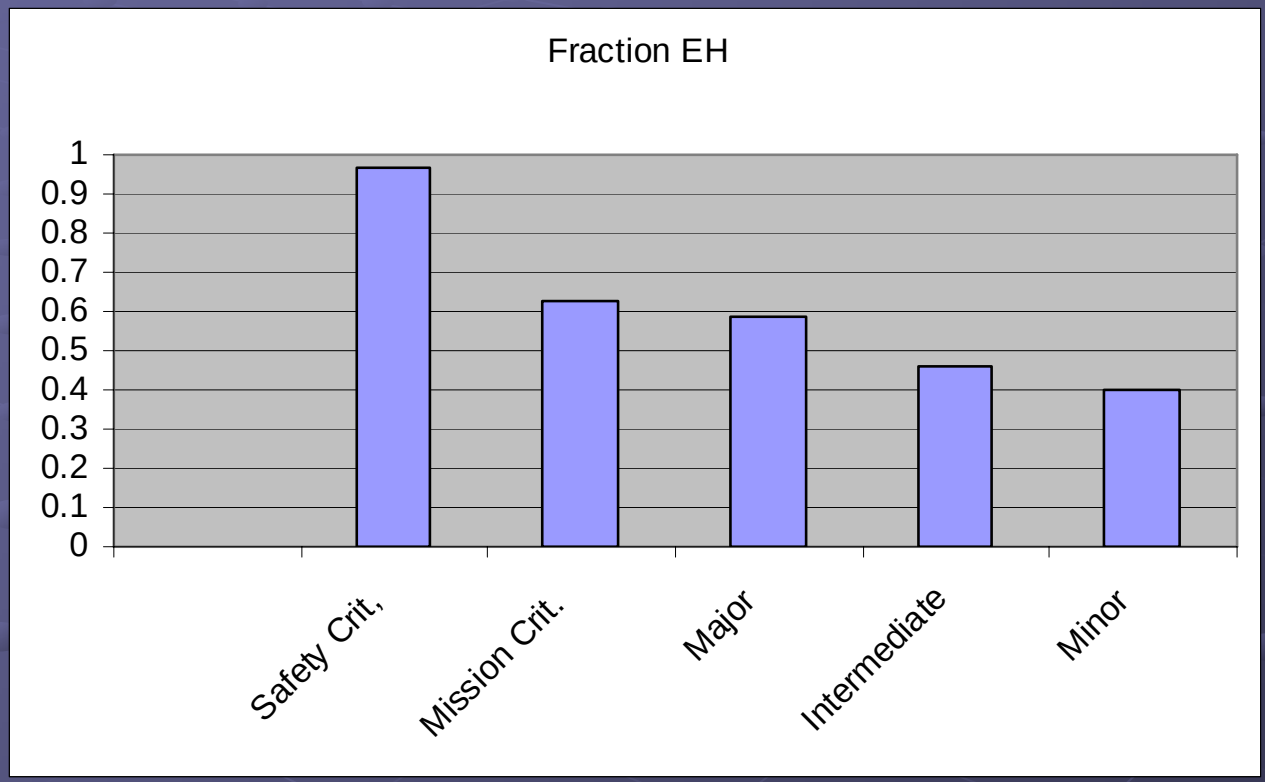


GLOBAL FAILURES



Kanoun, K. and T. Sabourin, “Software Dependability of a Telephone Switching System”, *Digest of Papers, FTCS-17,* Pittsburgh PA, July 1987, pp. 236 – 241

EXCEPTION HANDLING AND CRITICALITY



Hecht, H. and P. Crane, “Rare Conditions and their Effect on Software Failures”, *Proc. of the 1994 Annual Reliability and Maintainability Symposium”,* January 1994, pp. 334 – 337.

RELEVANT QUOTES

“The main line software code usually does its job. Breakdowns typically occur when the software exception code does not properly handle abnormal input or environmental conditions – or when an interface does not respond in the anticipated or desired manner.”

C. K. Hansen, *The Status of Reliability Engineering Technology 2001*, Newsletter of the IEEE Reliability Society, January 2001

“Therefore the identification and handling of the exceptional situations that might occur is often just as (un)reliable as human intuition.”

Flaviu Cristian “Exception Handling and Tolerance of Software Faults” in *Software Fault Tolerance,* Michael R. Lyu, ed., Wiley, New York, 1995

WHY THESE FAILURES?

- THE PROGRAMS WERE NOT TESTED UNDER THE CONDITIONS THAT CAUSED THE FAILURES
- THERE WERE NO REQUIREMENTS FOR TESTING UNDER THESE CONDITIONS
- GENERATING REQUIREMENTS FOR EXCEPTION HANDLING IS ***DIFFICULT***

WHY THE DIFFICULTY?

- EXCEPTION CONDITIONS ARISE FROM SEVERAL LEVELS
- EXCEPTION CONDITIONS ARE MORE DIFFICULT TO UNDERSTAND THAN MAIN LINE REQUIREMENTS
- EXCEPTIONS OCCUR INFREQUENTLY BUT REQUIRE DISPROPORTIONATE EFFORT

SOURCES OF EXCEPTIONS

OPERATIONAL REQUIREMENTS

LOSS OF POWER, COMMUNICATION, THERMAL CONTROL

IMPLEMENTATION DETAIL

CALIBRATION ANOMALIES, ACTUATOR STATES, OPERATOR INPUT

COMPUTING ENVIRONMENT

HARDWARE FAILURES, MEMORY ERRORS, EXECUTIVE, MIDDLEWARE

MONITORING AND SELF-TEST

OVER-TEMPERATURE SENSORS, SYSTEM PERFORMANCE TEST

APPLICATION SOFTWARE

ASSERTIONS, VIOLATION OF TIMING CONSTRAINTS, MODE CHANGES

WHO IS RESPONSIBLE?

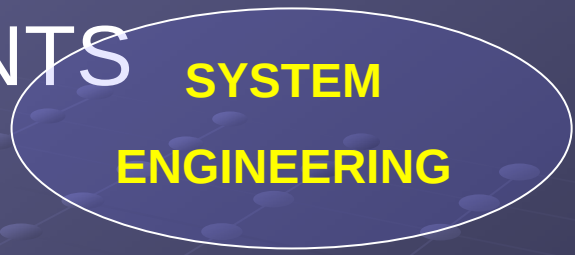
OPERATIONAL REQUIREMENTS

IMPLEMENTATION DETAILS

COMPUTING ENVIRONMENT

MONITORING AND SELF-TEST

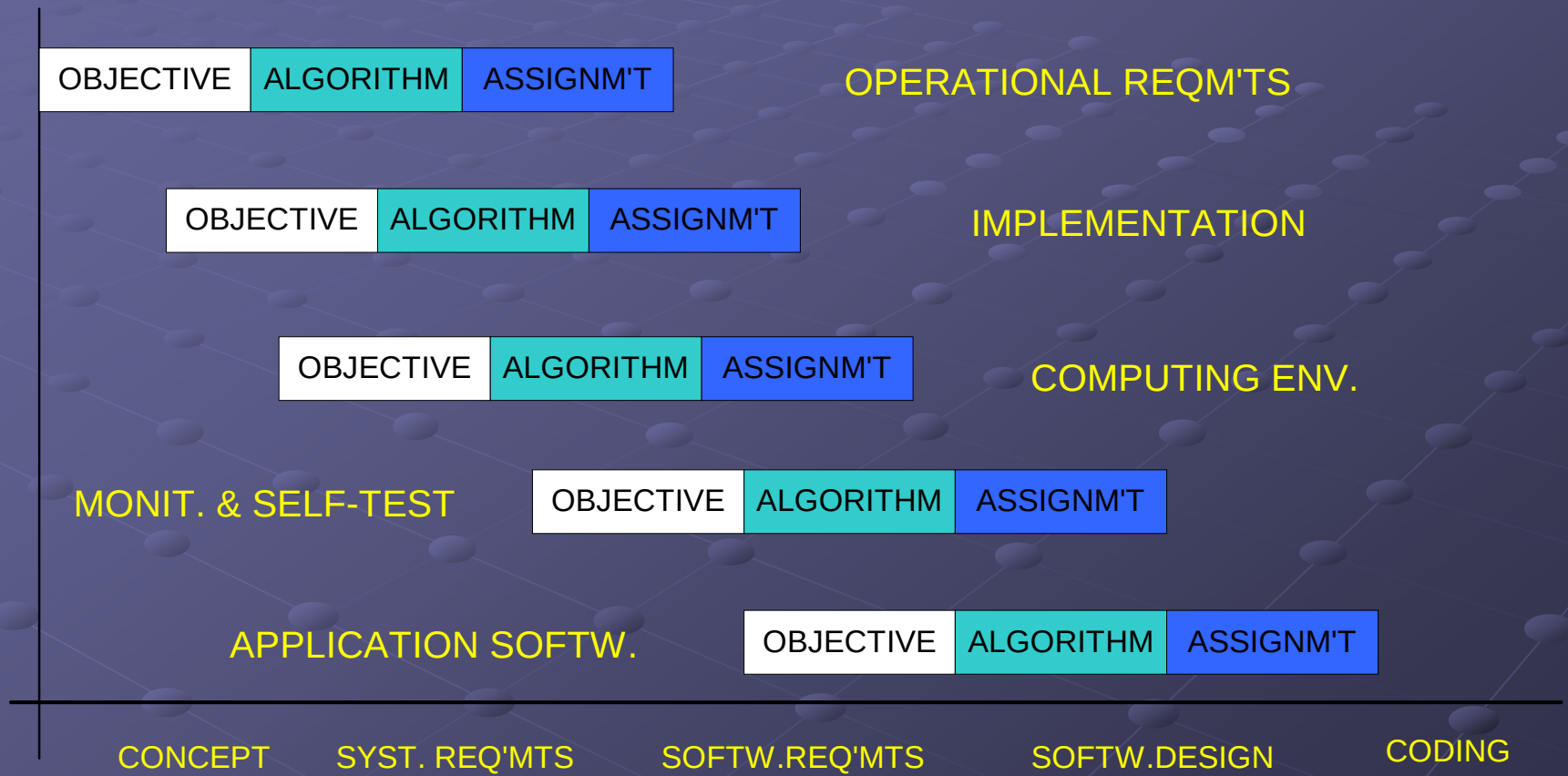
APPLICATION SOFTWARE



REQUIREMENT GENERATION

- OBJECTIVE
  - EXCEPTION CONDITION AND ACTION
- ALGORITHM
  - QUANTITATIVE CONDITION DESCRIPTION
  - TIMING AND RESPONSIBILITY FOR ACTION
- ASSIGNMENT
  - SPECIFY SOFTWARE IMPLEMENTATION OF ALGORITHM

DOES IT ADD UP?



BUILDING BLOCKS

- EXISTING PRACTICES
- EXPERIENCE
- TOOLS

- INTEREST GROUP
- WORKING GROUP
- RECOMMENDED PRACTICE

CONTACT

herb@sohar.com

Herb Hecht

310/338-0990 X110