EVALUATING THE AUTONOMY OF UAVs

Herbert Hecht, Ph. D.
SoHaR Incorporated
Culver City, California

NO HUMAN PILOT

- SAVES WEIGHT
- SIMPLIFIES DESIGN
- INCREASES LOSS ACCEPTANCE
- EXTENDS FLIGHT ENVELOPE

- ALL HANDLING OF ANOMALOUS CONDITIONS MUST BE PROGRAMMED AND TESTED IN ADVANCE

EXCEPTION HANDLING

SOFTWARE TESTING

REQUIREMENTS BASED

- ALL STATED REQUIREMENTS HAVE BEEN IMPLEMENTED
- EXECUTION PRODUCES DESIRED RESULTS

STRUCTURAL

- TRAVERSAL OF IMPLEMENTED PATHS PRODUCES NO UNDESIRABLE RESULTS

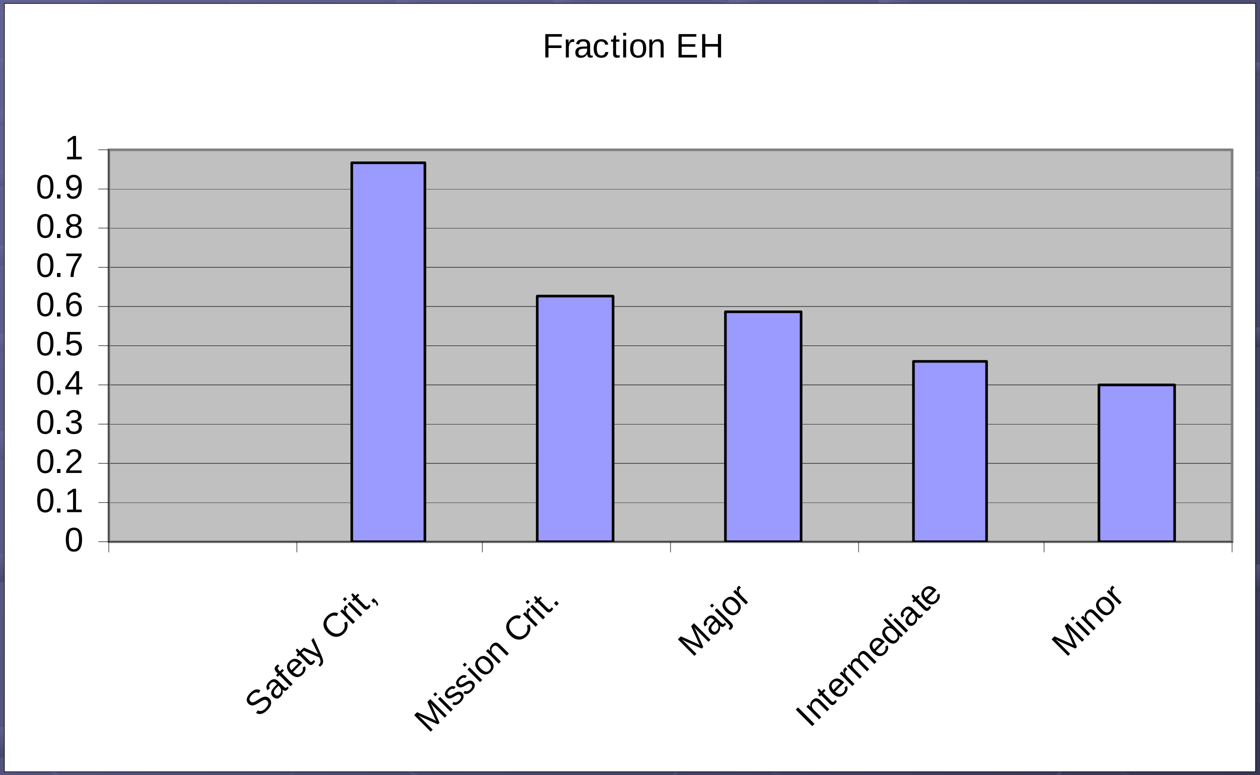
NEITHER APPROACH ASSURES ADEQUACY OF EXCEPTION HANDLING

EXCEPTION HANDLING

- VERY LITTLE LITERATURE
  - EXCEPT FOR LANGUAGE CONSTRUCTS
- NO GUIDANCE FOR SYSTEM LEVEL REQUIREMENTS FORMULATION
- MOST SOFTWARE FAILURES IN WELL-TESTED SYSTEM ARE DUE TO FAULTY EXCEPTION HANDLING

EXCEPTION HANDLING AND CRITICALITY

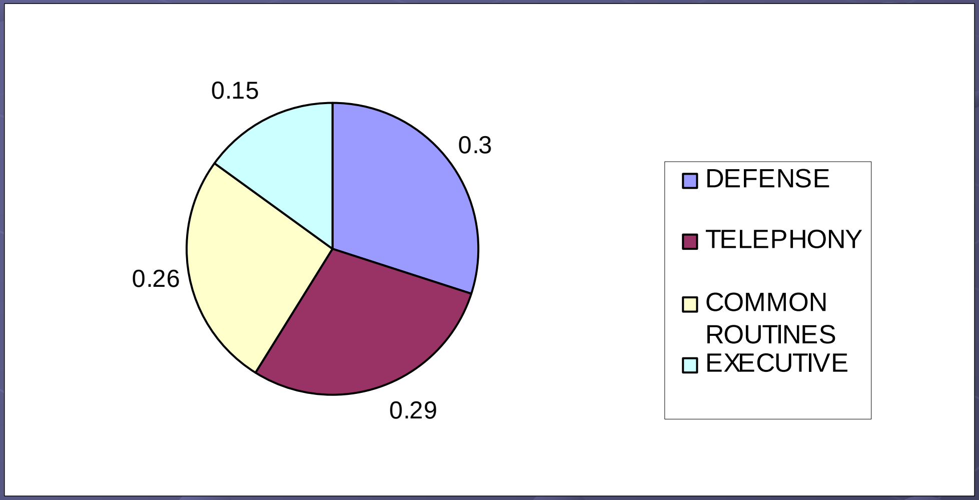
SPACE SHUTTLE AVIONICS SOFTWARE



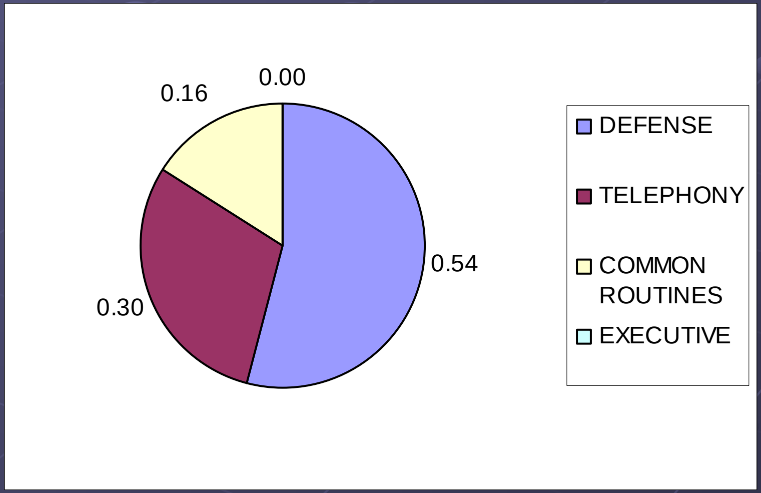
Hecht, H. and P. Crane, “Rare Conditions and their Effect on Software Failures”, *Proc. of the 1994 Annual Reliability and Maintainability Symposium”,* January 1994, pp. 334 – 337.

MORE EXCEPTION HANDLING FAILURES

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

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

SPECIFYING EXCEPTION HANDLING IS DIFFICULT

- EXCEPTION CONDITIONS ARISE FROM SEVERAL LEVELS

SPECIFYING EXCEPTION HANDLING IS DIFFICULT

- EXCEPTION CONDITIONS ARISE FROM SEVERAL LEVELS
- EXCEPTION CONDITIONS ARE MORE DIFFICULT TO UNDERSTAND THAN MAIN LINE REQUIREMENTS

SPECIFYING EXCEPTION HANDLING IS DIFFICULT

- 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 PROPULSION, ELECTRIC POWER, COMMUNICATION, THERMAL CONTROL

IMPLEMENTATION DETAIL

CALIBRATION ANOMALIES, ACTUATOR STATES, SENSOR 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

SYSTEM

ENGINEERING

IMPLEMENTATION DETAILS

EQUIPMEMT

SPECIALIST

COMPUTING ENVIRONMENT

VEHICLE

HEALTH MGM'T

MONITORING AND SELF-TEST

APPLICATION SOFTWARE

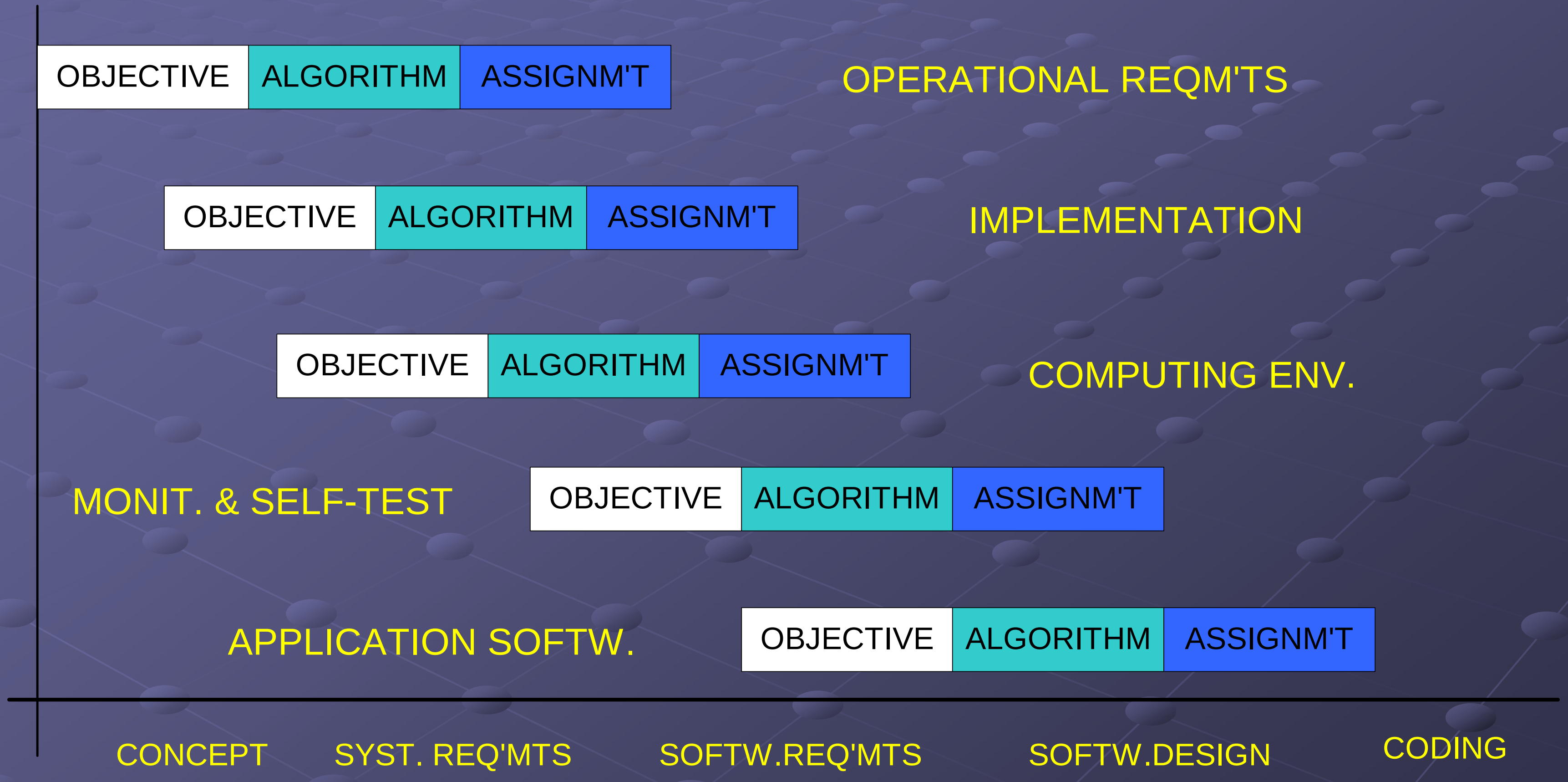
SOFTWARE

ENGINEERING

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?



OBJECTIVE ALGORITHM ASSIGNM'T

OPERATIONAL REQM'TS

OBJECTIVE ALGORITHM ASSIGNM'T

IMPLEMENTATION

OBJECTIVE ALGORITHM ASSIGNM'T

COMPUTING ENV.

MONIT. & SELF-TEST

OBJECTIVE ALGORITHM ASSIGNM'T

APPLICATION SOFTW.

OBJECTIVE ALGORITHM ASSIGNM'T

CONCEPT

SYST. REQ'MTS

SOFTW.REQ'MTS

SOFTW.DESIGN

CODING

SOLUTIONS TO THE PROBLEM

- SHARING EXISTING PRACTICES
- SHARING EXPERIENCE
- CREATING AND SHARING TOOLS

- INTEREST GROUP
- STANDARDS WORKING GROUP
- RECOMMENDED PRACTICE

HERBERT HECHT

herb@sohar.com

310.338.0990 X110