Current and Future Challenges of Software Reliability Assessment

October 30, 2003
US Army Conference on Applied Statistics
Napa Valley, CA

William Farr, PhD
B35, Combat Systems Branch Head
Phone: (540) 653-8388
Fax: (540) 653-8673
Email: farrwh@nswc.navy.mil

NAVSEA
DAHLGREN
Surface Warfare Center Division

Outline Of Presentation

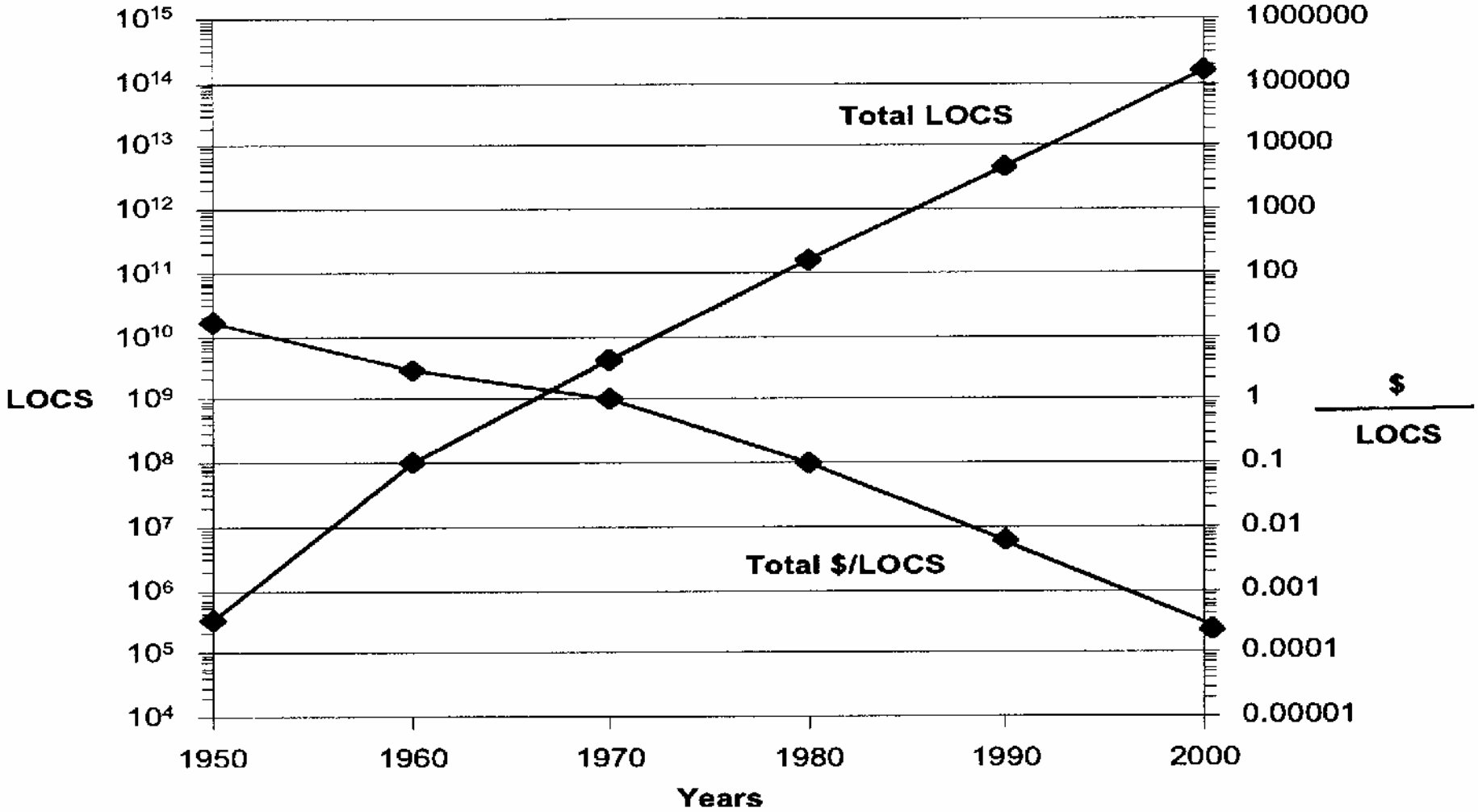
- Introduction
- What is Software Reliability and how is it assessed
- Current Challenges
- Future Challenges
- Summary

NAVSEA
DAHLGREN
Surface Warfare Center Division

Introduction

- Software is playing an ever greater role in our systems.
  - More intrinsically difficult tasks are now undertaken by software.
  - Software has moved from an auxiliary role to a primary role in providing critical services.
  - Software based applications are becoming an accepted part of our life.
- Software Reliability assessment is therefore of major concern to developers and users.

Lines of Code in Service: U.S. DoD



NAVSEA
DAHLGREN
Surface Warfare Center Division

What is Software Reliability and How is it Assessed

NAVSEA
DAHLGREN
Surface Warfare Center Division

DEFINITIONS

SOFTWARE RELIABILITY IS THE PROBABILITY THAT A GIVEN SOFTWARE PROGRAM WILL OPERATE WITHOUT FAILURE FOR A SPECIFIED TIME IN A SPECIFIED ENVIRONMENT.

SOFTWARE RELIABILITY ENGINEERING (SRE) IS THE APPLICATION OF STATISTICAL TECHNIQUES TO DATA COLLECTED DURING SYSTEM DEVELOPMENT AND OPERATION TO SPECIFY, PREDICT, ESTIMATE, AND ASSESS THE SOFTWARE RELIABILITY OF SOFTWARE-BASED SYSTEMS.

NAVSEA
DAHLGREN
Surface Warfare Center Division

Errors, Faults, Failures

- **ERROR** – Human action that results in software containing a fault.
- **FAULT** – A defect in code that can be the cause of one or more failures (synonymous with “bug”).
- **FAILURE** – The inability of a system or system component to perform a required function within specified limits. A departure of program operation from program requirements.

ERROR => FAULT(S) => FAILURE(S)

NAVSEA
DAHLGREN
Surface Warfare Center Division

Comparisons Between Software & Hardware

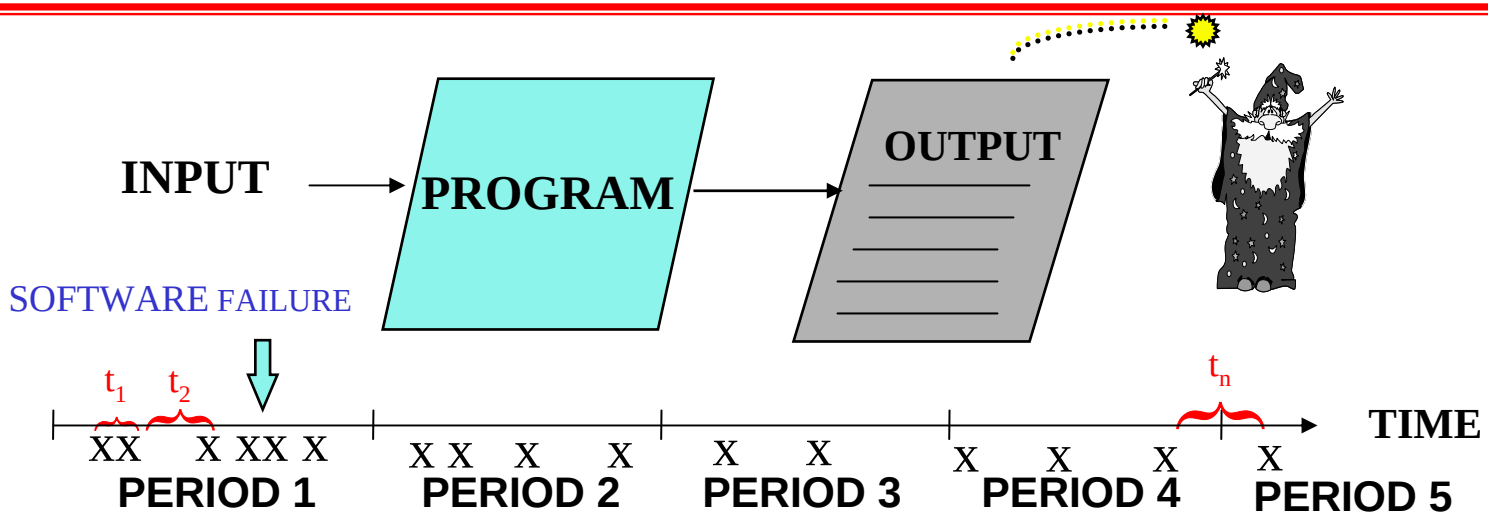
- Both are stochastic processes and can be described by probability distributions.
- Software faults are design faults not physical faults.
- Software does not “wear-out, burn-out, or deteriorate” over time.
- If there is a fault in a particular version of the software it will be in every copy.
- Software reliability is much more difficult measure to obtain and analyze.
- Software is continuously modified throughout its life cycle.

NAVSEA
DAHLGREN
Surface Warfare Center Division

APPROACHES OF S/W RELIABILITY ESTIMATION

- ERROR SEEDING/TAGGING MODELS
- DATA DOMAIN
- TIME DOMAIN

APPROACH TO ESTIMATING S/W RELIABILITY IN THE TIME DOMAIN



SOFTWARE RELIABILITY DATA:

NUMBER OF FAILURES/PERIOD (e.g., 6,4,2,3,1,...)

OR

TIME BETWEEN FAILURES (e.g., t_1, t_2, ... t_n)

NAVSEA
DAHLGREN
Surface Warfare Center Division

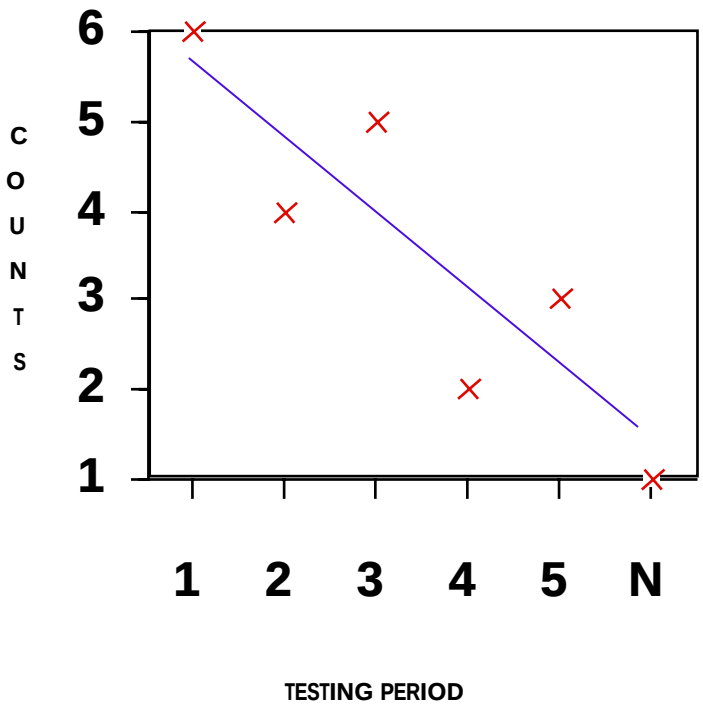
TIME UNITS

- "Counts" – by day, week, month, quarter, etc.
- "Time Between" – Wall Clock, CPU, Number of Test Cases, Natural unit (e.g. # transactions, # printed pages, etc.)

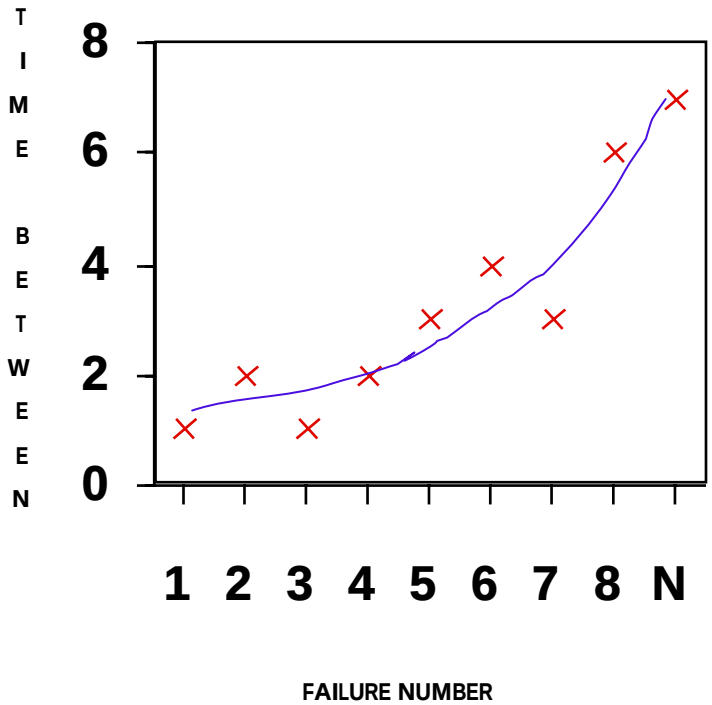
Personal Experience: Counts by month provided the easiest to implement without impacting prediction accuracy.

NAVSEA
DAHLGREN
Surface Warfare Center Division

SOFTWARE RELIABILITY MODELS IN THE TIME DOMAIN



FAILURE COUNTS PLOT



FAILURE TIME PLOT

NAVSEA
DAHLGREN
Surface Warfare Center Division

EXAMPLE SOFTWARE RELIABILITY MEASURES

ERROR COUNT MODELS

- TOTAL NUMBER OF ERRORS
- EXPECTED NUMBER OF ERRORS IN FUTURE TESTING PERIODS
- FAILURE RATE
- TESTING TIME REQUIRED TO ELIMINATE THE NEXT K ERRORS OR TO ACHIEVE A SPECIFIED FAILURE RATE
- CURRENT PROGRAM RELIABILITY

TIME BETWEEN MODELS

- TOTAL NUMBER OF ERRORS
- MEAN TIME TO NEXT FAILURE
- FAILURE RATE
- TESTING TIME REQUIRED TO ACHIEVE A SPECIFIED RATE
- CURRENT PROGRAM RELIABILITY

NAVSEA DAHLGREN Surface Warfare Center Division

EXAMPLES OF SOFTWARE RELIABILITY MODELS

Time Models

- Littlewood & Verrall Bayesian Model
- Musa Basic Execution Model
- Geometric Model*
- Non-homogeneous Poisson Process Model
- Musa Logarithmic Poisson Model*
- Jelinski-Moranda Model

Error Count Models

- Generalized Poisson Process Model
- Non-homogeneous Poisson Process Model for Counts
- Brooks and Motley's Model
- Schneidewind Model*
- S-shaped Reliability Growth Model*

Note: The above models are all in SMERFS^3 and CASRE.

The models with the "*" have shown themselves to be especially applicable to many data sets.

NAVSEA
DAHLGREN
Surface Warfare Center Division

CURRENT STATUS OF RELIABILITY MODELING

- Currently over 100 models in the Literature
- SRE Newsletter, International Symposium (14 held – next in Denver), Numerous Publications on the subject including 4 key books, (John Musa – “Software Reliability Engineering”)
- Standards already exist, e.g. IEEE 982.1 and 982.2 and AIAA’s ANSI/AIAA “Recommended Practice for Software Reliability” (R-013-1992)
- Software Reliability modeling is applied in a wide variety of applications, e.g. Communications (AT&T Best Practice), Space Program (Shuttle, Galileo); and the DOD

NAVSEA
DAHLGREN
Surface Warfare Center Division

Current Challenges

- **Diversity of Reliability Concerns**
- **Ultra reliability requirements (<=10^{-8}) of some software systems**
- **Demonstrated value of software reliability assessment**
- **Basic Assumptions**
  - **Independence, fault correction, & the environment**
- **Changing nature of software development**
  - **Component-based multi-tiered software; COTS; quality standards; CMMI; web development; growth of programming languages; programmers without formal training**
  - **Emphasis on shorter development time; a push to market**

NAVSEA
DAHLGREN
Surface Warfare Center Division

Future Challenges

Min Xie – "Software reliability is ready for a new phase of development where emphasis is on practical implementation."

- **Specify and write high quality, modular, high performance software for a wide variety of applications**
- **Combining various views of reliability (testing, code inspections, audits, reliability modeling, etc.) into an overall assessment**
- **Develop techniques for early reliability assessment**
- **Develop reliability assessment for evolving technologies (Example: autonomous agents)**
- **Develop reliable software for web based systems**
  - **Support 24/7 applications**
- **Demonstrate ultra-high reliability for safety critical systems**
- **Develop reliability models for complex systems**
  - **System of systems**
  - **Hardware & software**
  - **Factor the human element into the equation**

NAVSEA
DAHLGREN
Surface Warfare Center Division

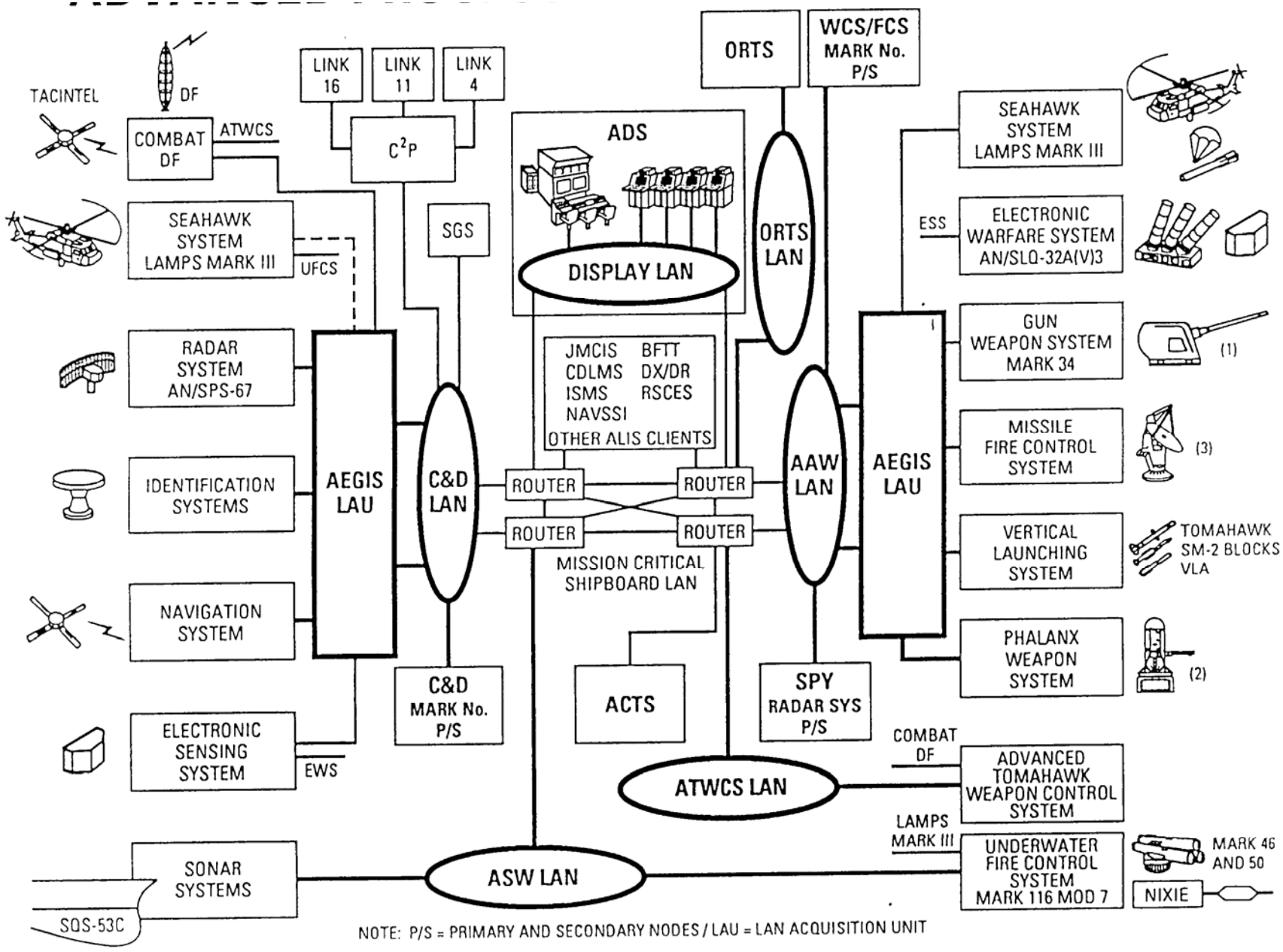
Examples of Complex Systems

- Air Traffic Control System
- Baggage Handling System at the Denver Airport
- Space Shuttle
- Medical Test Equipment
- Banking & E-commerce
- Navy warship

NAVSEA
DAHLGREN
Surface Warfare Center Division

AEGIS DDG

WARSHIP ("SYSTEM of SYSTEMS")



NAVSEA
DAHLGREN
Surface Warfare Center Division

Human Factors Considerations on Reliability

NAVSEA
DAHLGREN
Surface Warfare Center Division

Do's and Don'ts of Software Reliability

Do's

1. Consider it as a quantitative measure to be used in determination of software quality.
2. Track reliability over time and/or software version to determine whether reengineering is needed.
3. Develop specific objectives for a software reliability effort.
4. Develop an automated data collection process and database.

Don't

1. Consider one software reliability model is applicable for all situations.
2. Extrapolate results beyond the environment in which the data is generated.
3. Consider it as the sole data point upon which to make judgments.

NAVSEA
DAHLGREN
Surface Warfare Center Division

SUMMARY

- **RELIABILITY ANALYSIS CAN PROVIDE USEFUL INFORMATION TO ASSESS BOTH THE PRODUCT AND THE ORGANIZATION'S DEVELOPMENT PROCESS**
- **Challenge: WHEN MAKING AN OVERALL PRODUCT ASSESSMENT, CONSIDER RELIABILITY MODELING AS PROVIDING ONE PERSPECTIVE - OTHERS NEED TO BE CONSIDERED AS WELL**
- **Challenge: THERE IS A NEED TO QUANTIFY BOTH COMPONENT (HARDWARE & SOFTWARE) AND SYSTEM'S RELIABILITY**
- **Challenge: FOR TOTAL SYSTEM'S RELIABILITY THE HUMAN FACTOR MUST BE WORKED INTO THE FRAMEWORK ALSO**
- **Challenge: RELIABILITY NEEDS TO BE CONSIDERED OVER THE ENTIRE LIFECYCLE OF THE SYSTEM**
- **Challenge: NEED TO SHARE INFORMATION ON APPLYING THE METHODOLOGY TO DEMONSTRATE ROI**

NAVSEA
DAHLGREN
Surface Warfare Center Division

KEY REFERENCES

- *Software Reliability Engineering*, by John Musa, McGraw-Hill, 1999.
- *Software Reliability Measurement, Prediction, Application*, by J. Musa, A. Iannino, and K. Okumoto, McGraw-Hill, 1996.
- *Software Reliability Modeling*, by M. Xie, World Scientific, 1991.
- *Handbook of Software Reliability Engineering*, Edited by Michael Lyu, McGraw-Hill, 1997.
- *Recommended Practice for Software Reliability*, ANSI/AIAA, R-013-1992, ISBN 1-56347-024-1
- Center For Software Reliability - http://www.csr.city.ac.uk/
- Links to Others - http://members.aol.com/JohnDMusa/
  http://www.dacs.dtic.mil/