THE OHIO STATE UNIVERSITY

SOFTWARE RELIABILITY

Challenges and Directions

Carol Smidts
Department of Mechanical and Aerospace Engineering
The Ohio State University
Smidts.1@osu.edu

To be presented at the
Reliability Engineering 25th Anniversary Symposium
April 2, 2014
University of Maryland, College Park

CHARACTERISTICS IN CONTRAST

MORE RECENT

- First hardware reliability paper appears in 1952 in Proceedings of the Institute of Radio Engineers.
- First software reliability paper appears in 1975 in IEEE Transactions on Software Engineering.

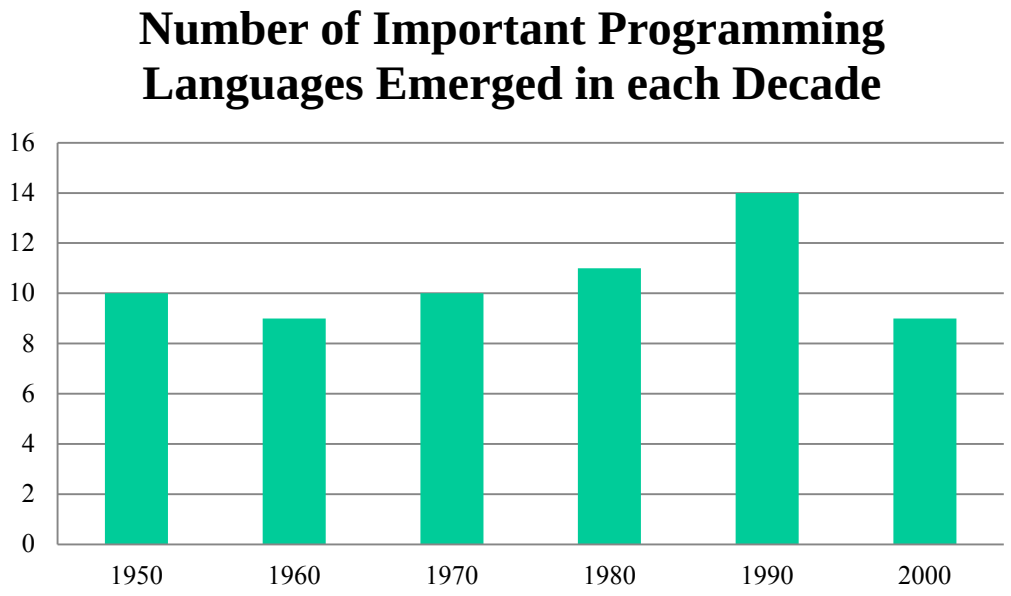
MORE COMPLEX

- The complexity of typical hardware systems is several hundreds of components (e.g., nuclear power plants).
- The complexity of current software systems is millions of lines of source code (e.g., 15 millions for the Linux kernel). Assuming a typical function consists of 200 lines of code, there are approximately 75,000 functions in the Linux kernel.

CHARACTERISTICS IN CONTRAST

EVOLVES EXTREMELY FAST

- The number of important programming languages introduced per decade is approximately 10. This number has been constant since 1950.



CHARACTERISTICS IN CONTRAST

EVOLVES EXTREMELY FAST (Cont'd)

- Programming paradigms have changed from non-structured to structured, procedural to object-oriented.
- Six main paradigms currently coexist: imperative, declarative, functional, object-oriented, logic and symbolic.

ALWAYS TIED TO HARDWARE

- Software does not run in isolation
- Software is tied to a computer platform
- As such failures are never observed in isolation
- This has led some to not want software to be modeled at all

CHARACTERISTICS IN CONTRAST

DIFFERENT FAILURE MODE

- Hardware:
  - Hardware wears out leading to degraded performance
  - Failures are triggered due to harsh environment like excess heat and radiation
- Software:
  - Software does not wear out
  - Failures are due to latent faults that are triggered and propagate into failures

HIGHLY DEPENDENT UPON ITS ENVIRONMENT

- Software is particularly sensitive to the environment

CONTINUITY ASSUMPTION ONLY VALID WITHIN THE CONFINES OF A LARGE NUMBER OF SMALL SUBDOMAINS

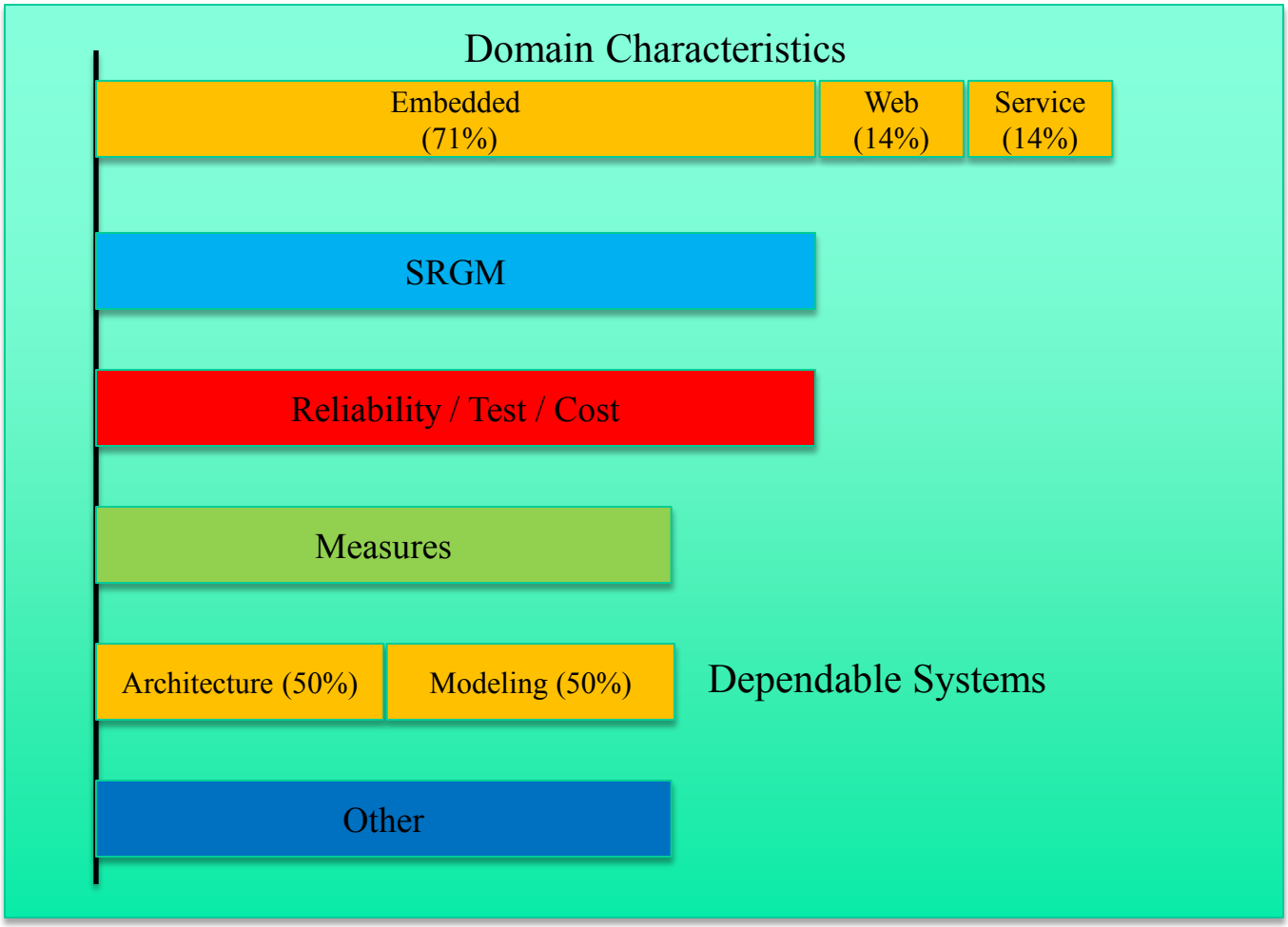
- Predicates create non continuous behavior in program logic.
- The typical ratio of predicates over lines of code is at the magnitude of 1/10.

ONE OF A KIND

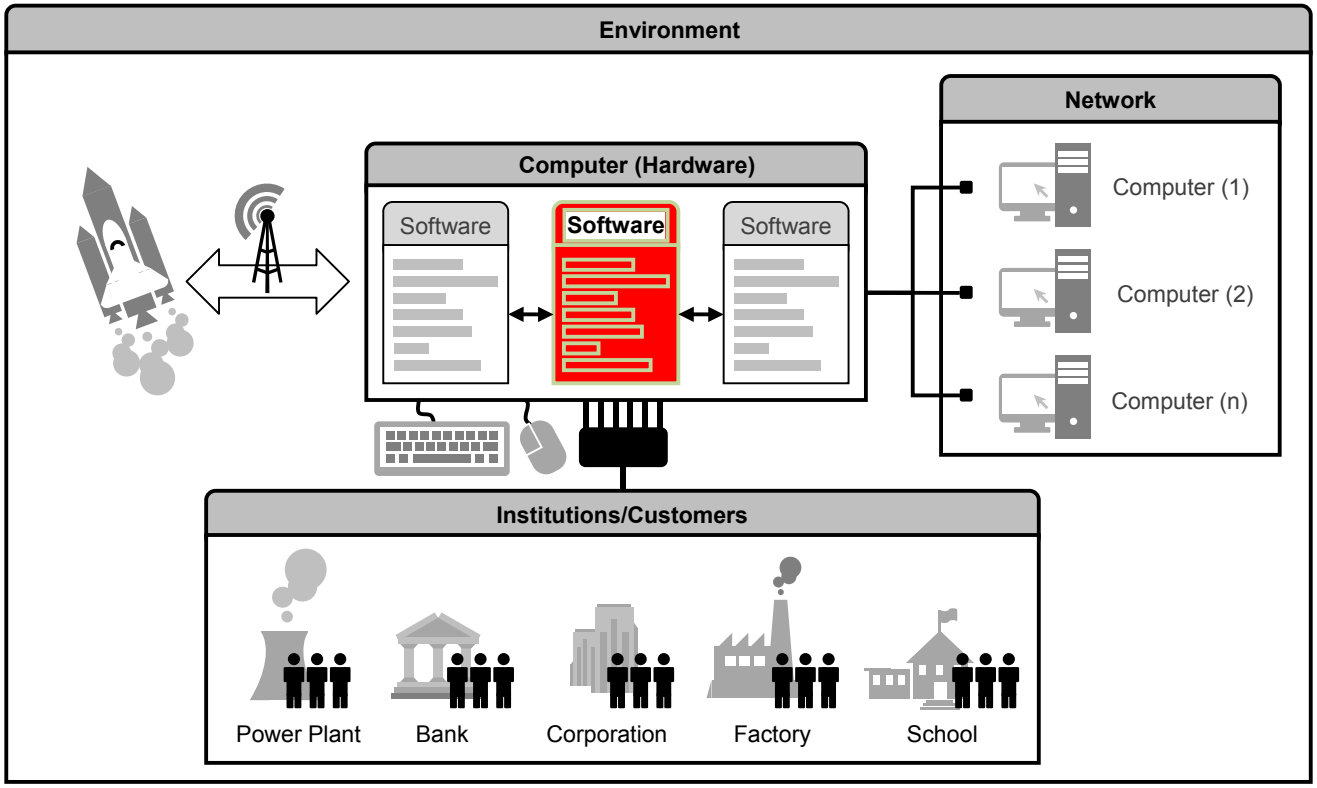
- Data is difficult to collect

CURRENT AREAS OF RESEARCH

Based on a review of papers published between 2008-2013 in the Proceedings of the International Symposium on Software Reliability Engineering (ISSRE) [excludes 2012]

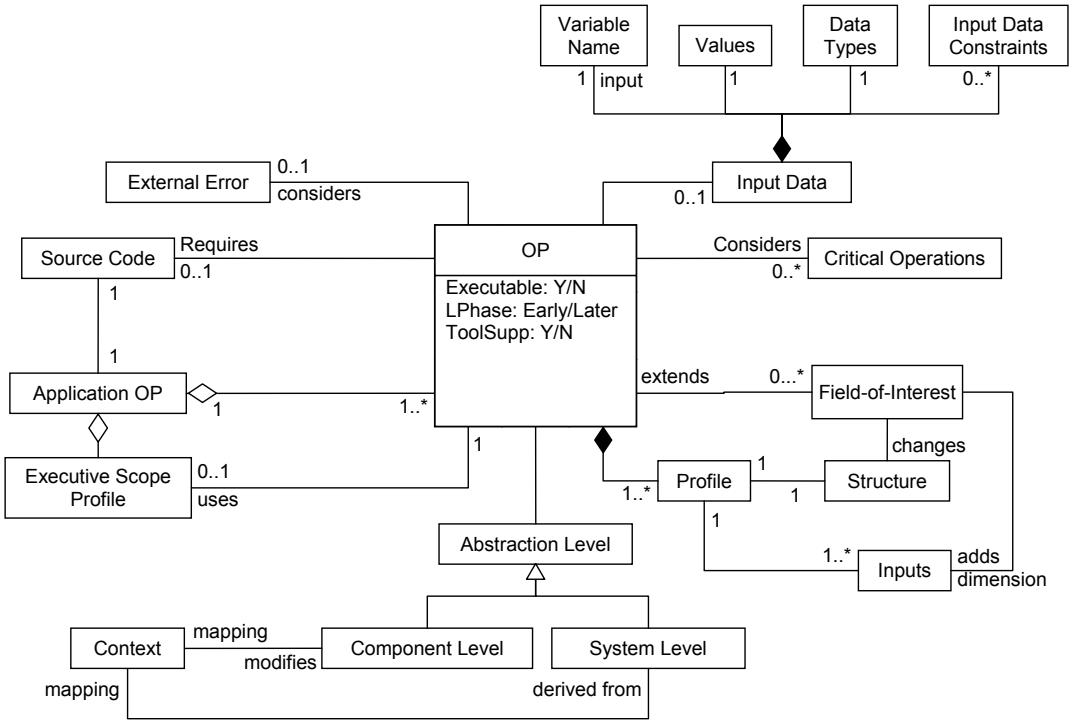


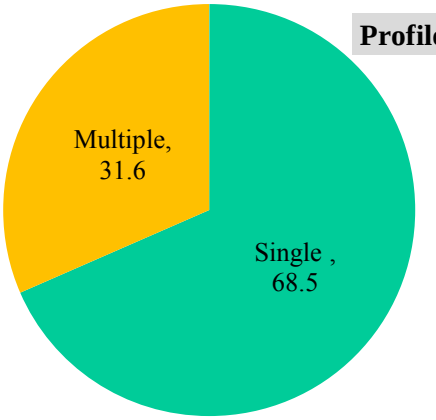
AREAS OF RESEARCH EXAMPLES: OP DEFINITION

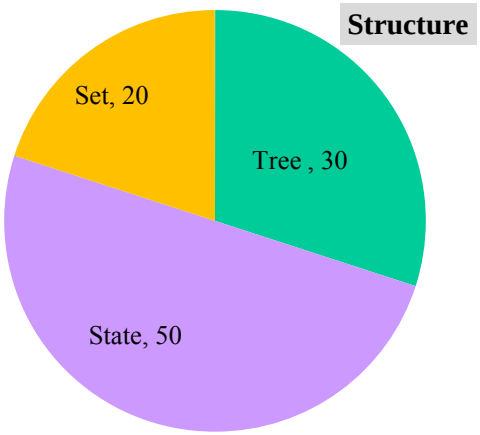


Extract from: Smidts, C., Mutha, C., Rodríguez, M., & Gerber, M. J. (2014). Software testing with an operational profile: OP definition. *ACM Computing Surveys (CSUR)*, *46*(3), 39.

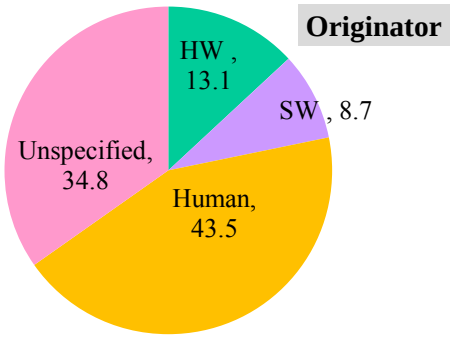
AREAS OF RESEARCH EXAMPLES: OP DEFINITION

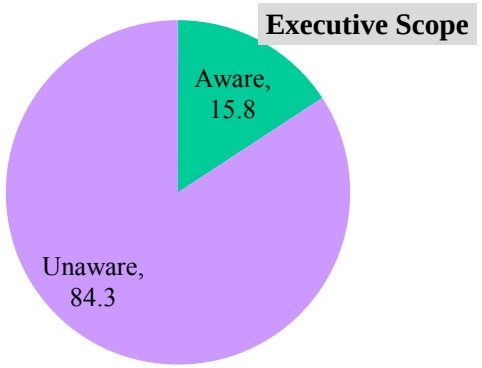


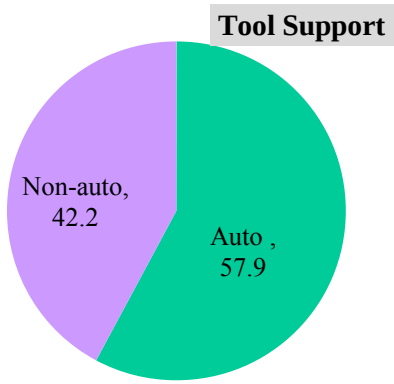


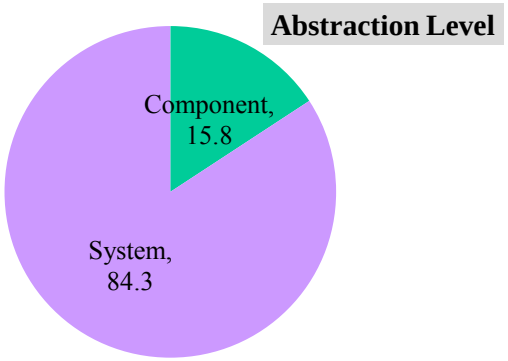


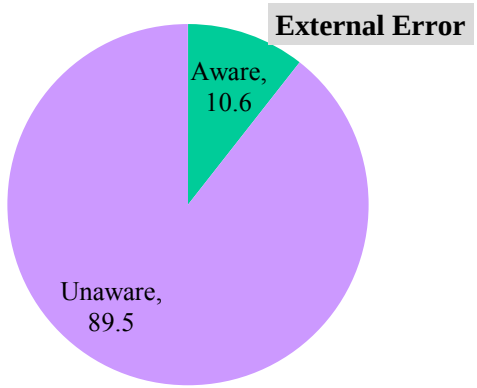
Extract from Smidts, C., Mutha, C., Rodríguez, M., & Gerber, M. J. (2014). Software testing with an operational profile: OP definition. *ACM Computing Surveys (CSUR)*, *46*(3), 39.

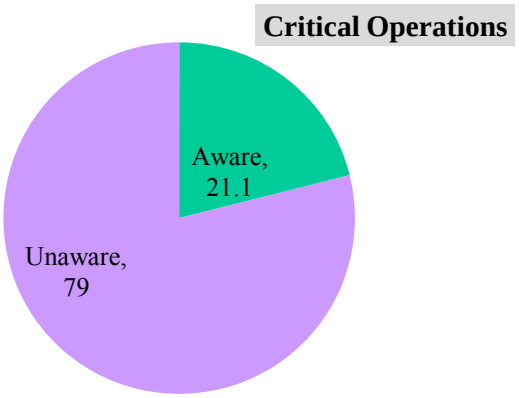

Originator
HW , 13.1
SW , 8.7
Human, 43.5
Unspecified, 34.8

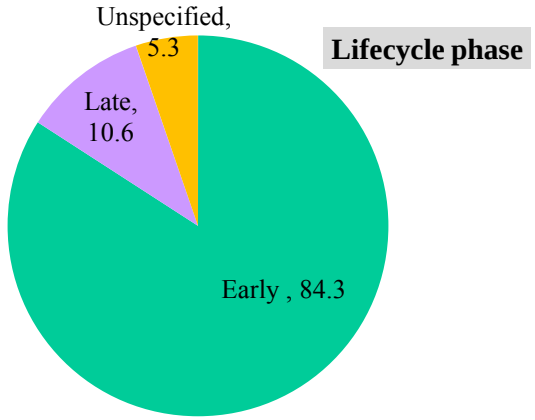

Executive Scope
Aware, 15.8
Unaware, 84.3


Tool Support
Non-auto, 42.2
Auto , 57.9

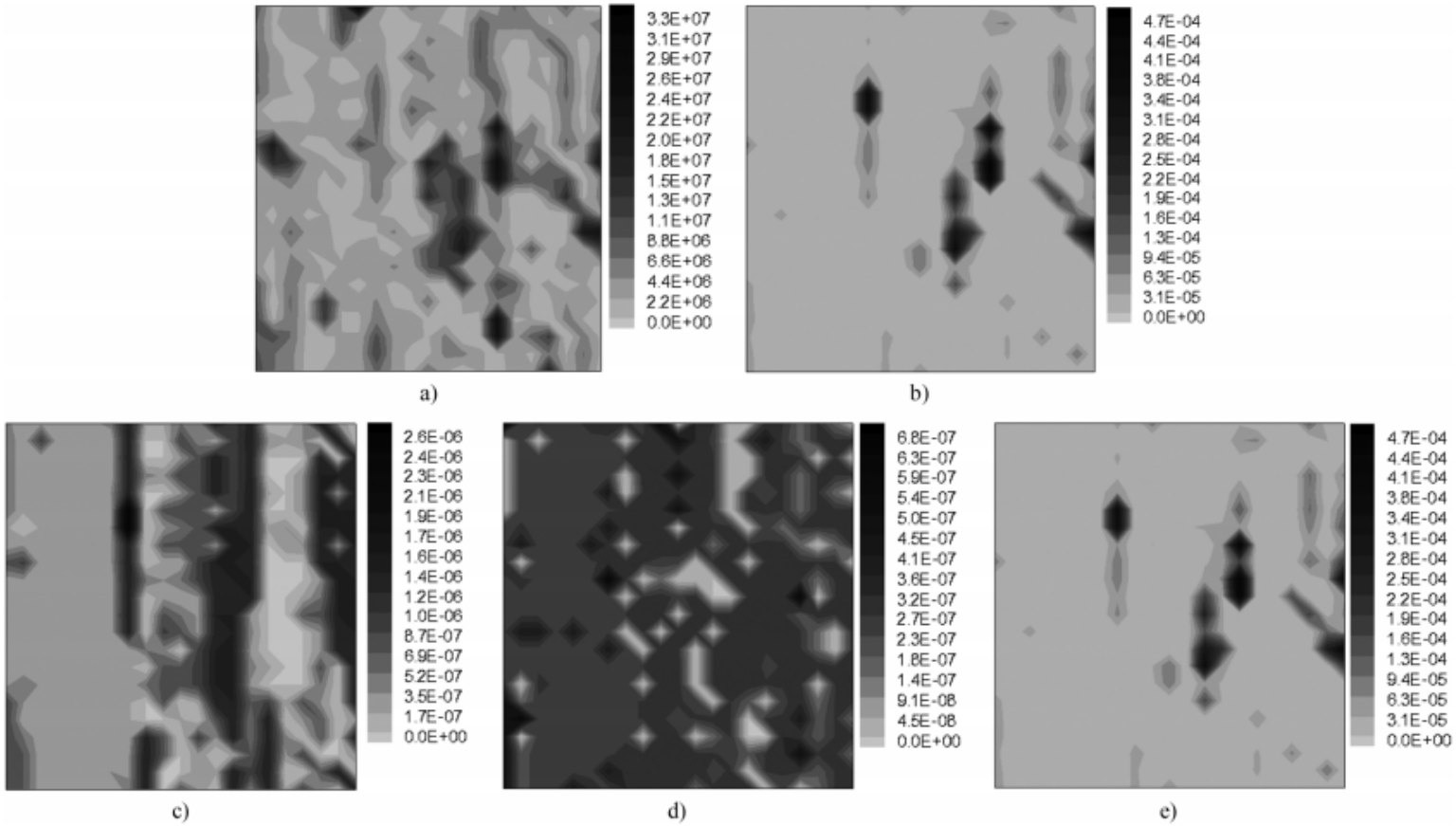

Abstraction Level
Component, 15.8
System, 84.3


External Error
Aware, 10.6
Unaware, 89.5


Critical Operations
Aware, 21.1
Unaware, 79


Unspecified, 5.3
Lifecycle phase
Late, 10.6
Early , 84.3

AREAS OF RESEARCH EXAMPLES: SOFTWARE AND HARDWARE RELIABILITY



ALU maps, showing usage and probability profiles. (a) Usage in terms of number of demands. (b) Delay probability profile. (c) Different-Function probability profile. (d) Stuck-at probability profile. (e) Combined failure probability profile.

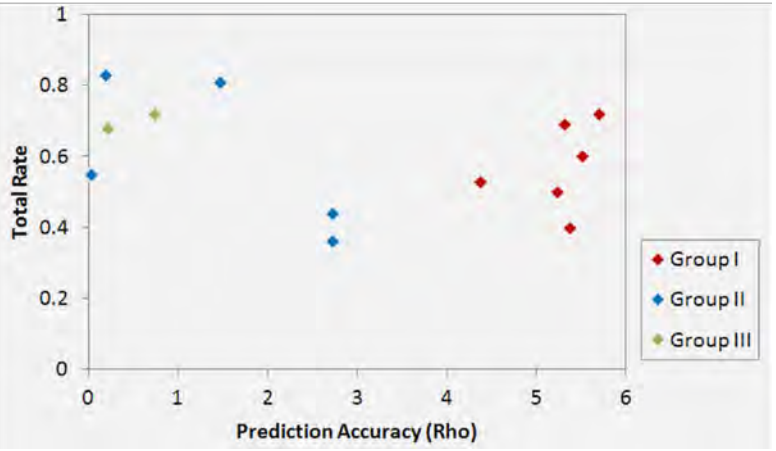
Extracted from: Bing H.; Rodriguez, M.; Ming Li; Bernstein, J.B.; Smidts, C.S., "Hardware Error Likelihood Induced by the Operation of Software," *Reliability, IEEE Transactions on* , vol.60, no.3, pp.622,639, Sept. 2011

AREAS OF RESEARCH EXAMPLES: SOFTWARE MEASURES TO SOFTWARE RELIABILITY

Group	Root Metric	Total Rate	Rank	Inaccuracy Ratio
I	BLOC	0.4	L	5.3764
	CMM	0.6	M	5.5091
	CC	0.72	H	5.6927
	FP	0.5	L	5.2303
	RSCR	0.69	M	5.3095
	SDC	0.53	M	4.3765
II	CEG	0.44	L	2.7243
	CF	0.81	H	1.4662
	COM	0.36	L	2.7211
	DD	0.83	H	0.1853
	RT	0.55	M	0.0334
III	FD	0.72	H	0.7397
	TC	0.68	M	0.2146

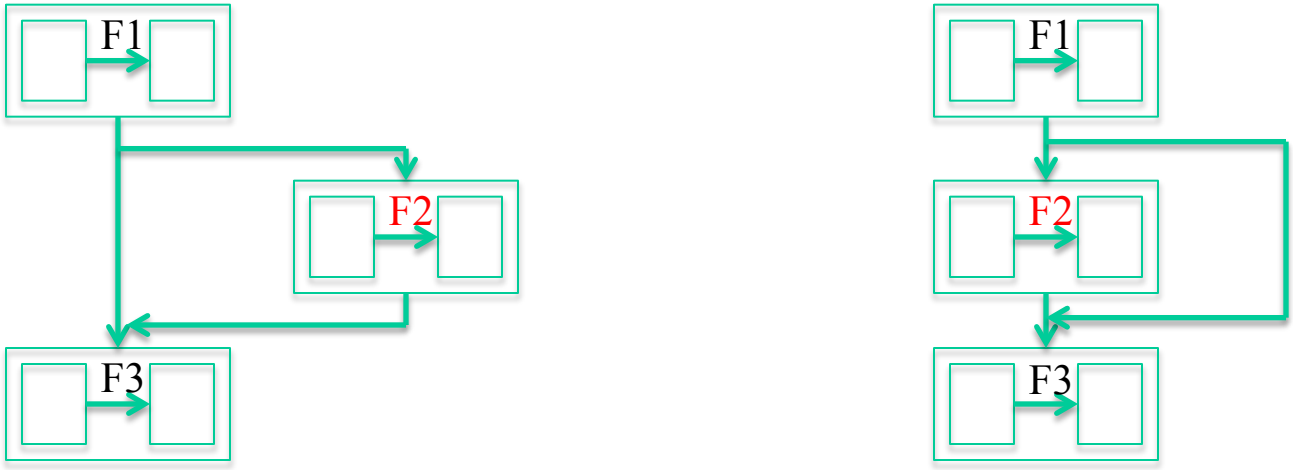
Inaccuraccy ~ Group + Strata + Group*Strata

	Sum Sq
Group	54.556
Strata	3.986
Group:Strata	2.424
Residuals	1.901



AREAS OF RESEARCH EXAMPLES: CHARACTERIZING SOFTWARE FAILURE MECHANISMS

#	Defect Name
1	Missing function
2	Extra function
	…



NEW AREAS OF RESEARCH

When software is mobile	Cloud, distributed, multi-core
When software reinvents itself	Adaptive systems
Different hardware platforms	Multi-core, quantum
Seeing software development as a human activity	Human reliability as way to study software reliability
Conquering complexity	Multi-scale modeling
Software reliability in relation to other attributes	Defining the relations between reliability, security, safety, …
The safety case	What has software reliability to say
Human and Software Interaction	Modeling the reciprocal relationship between software and humans
When software is hardware	FPGAs

THANK YOU

QUESTIONS?