29th Annual INCOSE international symposium

Orlando, FL, USA

July 20 - 25, 2019

Sarah Sheard, Mike Pafford, Mike Phillips

Systems–Software Engineering Role Interface

www.incose.org/symp2019

This material is based upon work funded and supported by the Department of Defense under Contract No. FA8702-15-D-0002 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

DM19-0630

"Systems Engineering-Software Engineering Interface for Cyber-Physical Systems"

S. Sheard, M. Pafford, M. Phillips

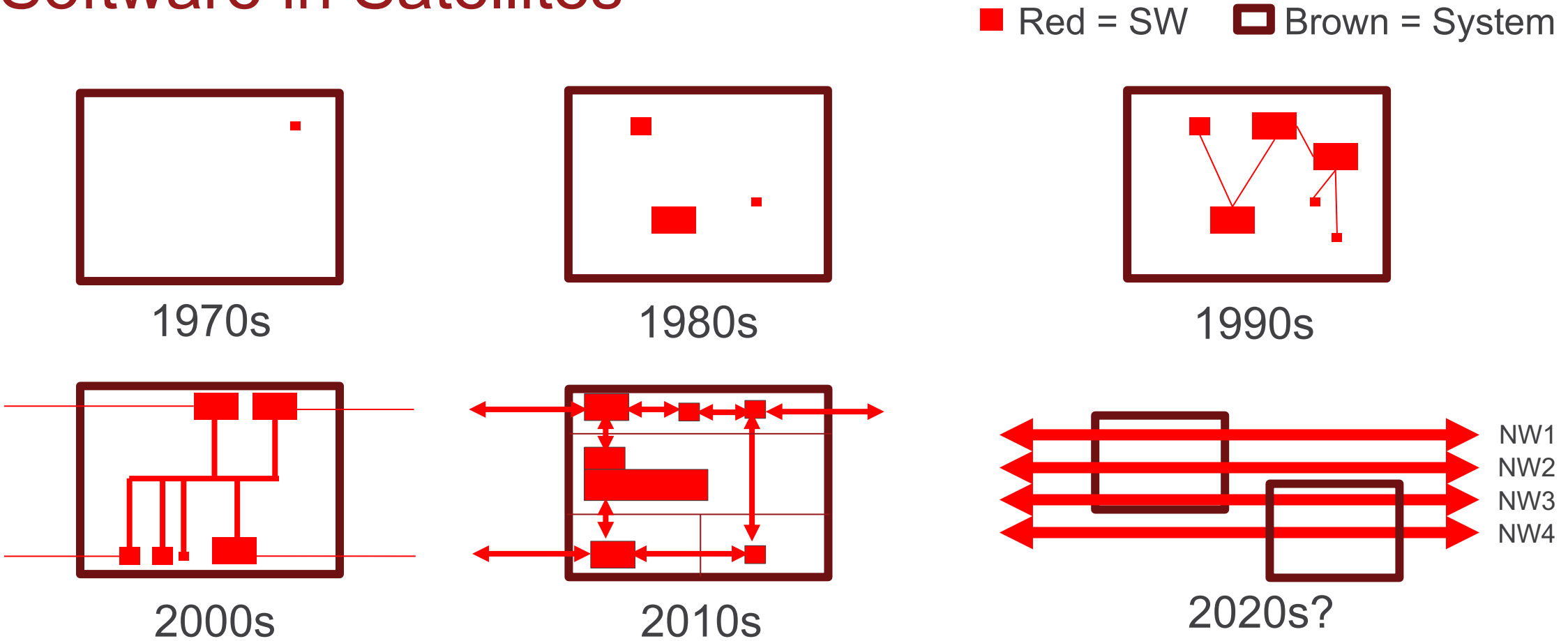
INCOSE's **"Systems and Software Interface Working Group"** (SaSIWG)

Agenda

- How are CPS different?
- SysE and SWE roles and activities
  - Including agile roles and processes
- Questions to ask
- Vision

*Paper gives more history and recounts failures of several safety-critical systems involving software, and more on SoS roles

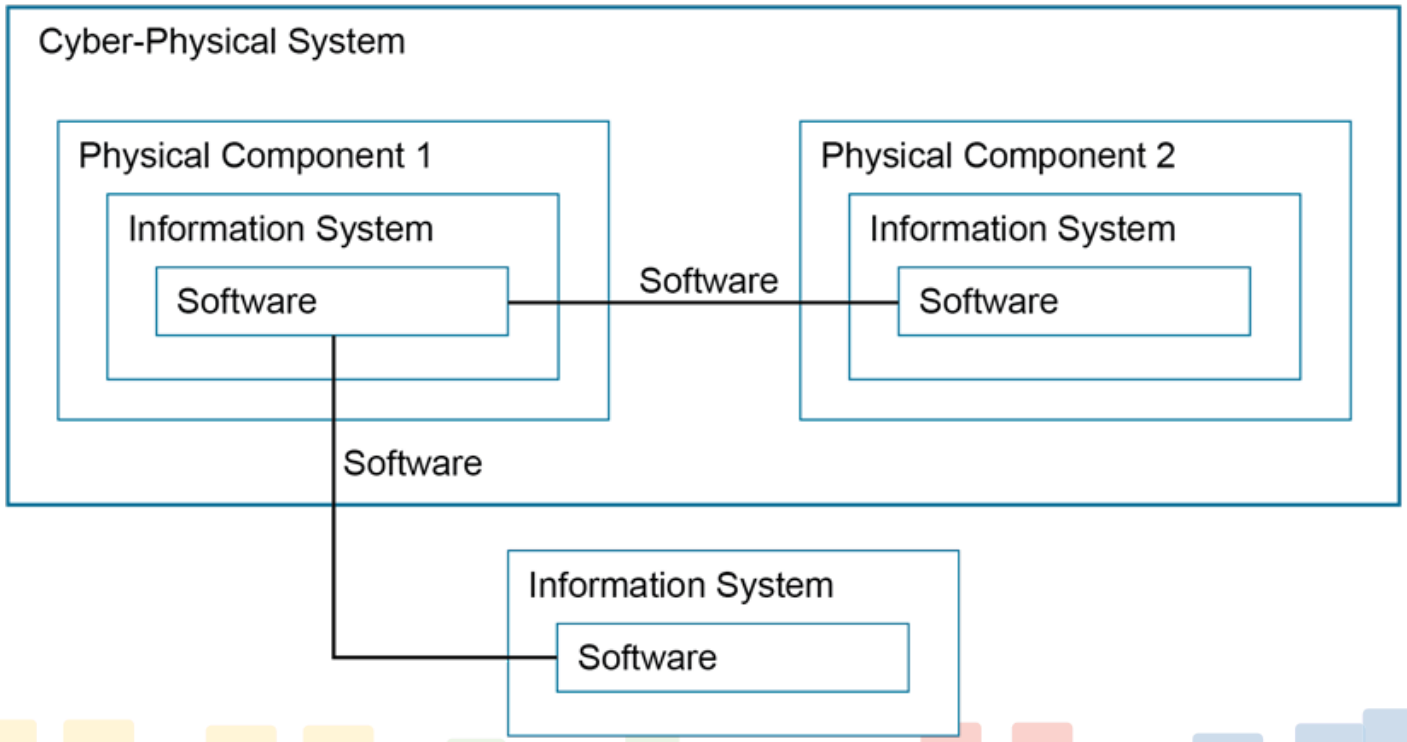
Software in Satellites



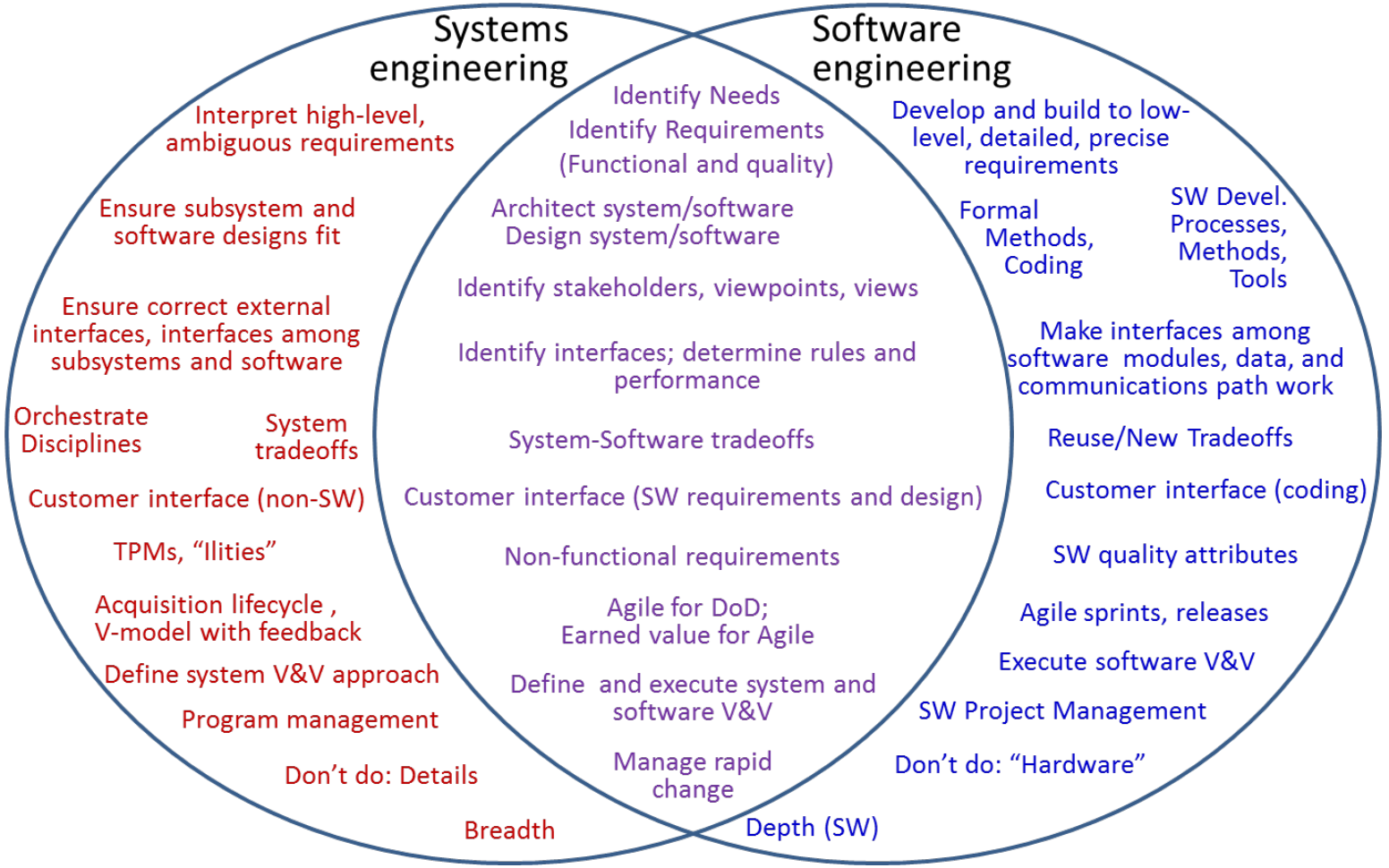
Reference: Sheard, 2014. "The changing relationship of systems and software in satellites," July, SEI Blog: https://insights.sei.cmu.edu/sei_blog/2014/07/the-changing-relationship-of-systems-and-software-in-satellites-a-case-study.html

New Kinds of Systems

- “Systems of systems”
- “Complex systems” (“Emergent systems”)
- “Cyber-physical systems”



SysE and SWE Responsibilities



Both: Perform coordinated roles (next 3 slides)

SWE: Architect, design, and implement SW in a system context

SysE: Adapt SysE practices to include SWE as important participants

Reference: Sheard, 2014. "Needed: Improved Collaboration Between Software and Systems Engineering," May, SEI Blog: https://insights.sei.cmu.edu/sei_blog/2014/05/needed-improved-collaboration-between-software-and-systems-engineering.html

Tasks, Roles, and Activities[1]

Tasks	Systems Engineering	Software Engineering
1. Implement	(none)	Programmer, Coder, Tester of code, Agile team roles (coding), Debugging and documentation, Maintain skills
2. Architect/ Design: ID components & interfaces, alloc. rqts	**SD** System Designer, System Architect, **RO** Requirements Owner, **G** Glue among subsystems Stay faithful to, or deliberately modify, design during maintenance	Design software architecture Architect software, including detailed design. Agile team roles (refactoring). Keep design/architecture in mind during maintenance
4. Analyze System and own external interfaces	Analyze budgets, margins, timing, failure modes Characterize external systems & interfaces; **SA** System Analyst; **CI** Customer Interface	Analyze budgets, margins, timing, failure modes Characterize external systems & interfaces; architecture analysis Maintenance analyses

1- and 2-letter roles: from (Sheard, 1996) “Twelve Systems Engineering Roles”

Tasks, Roles, and Activities$_2$

Tasks	Systems Engineering	Software Engineering
3. Lead & Coordinate (technical)	Liaison to other disciplines incl. SWE and component builders Risk identification and balancing **CO** Coordinator	Software risk identification and escalation to system risk as approp. Liaison with other SWEs and with SysE Agile team roles (leading, coordinating)
5. V&V	Plan and monitor system test processes and results Validate requirements through system operation **VV** V&V Engr.; **LO** Logistics/Ops Engr	Architecture evaluation Plan and execute SW V&V throughout build Agile team roles (adequacy, need vs backlog) Maintenance V&V
6. Manage people	Manage systems engineers, ensure they can learn broadly **TM** Technical Manager **PE** Process engineer	Manage software engineers, ensure they can keep current

Tasks, Roles, and Activities$_3$

Tasks	Systems Engineering	Software Engineering
7.Coordinate (project, mgmt.)	Interact with system-level customers; maintain agreements and resources; **CO** Coordinator	Interact with users and software-proficient customers Obtain agreements and resources
8. Plan and monitor	Technical management **TM** Technical manager, **PE** Process Engineer	SW Task or sprint management Agile sponsor roles
9. Manage risk	Balance application of resources to reduce and mitigate system-level risks	Apply resources for software risk mitigation; escalate SW-related risks to system level as needed
10. Manage configurations, data, and quality	**IM** Do CM, QA, and data manage-ment for system as a whole; maybe for non-software-related pieces Identify system quality measures and measure & improve system quality	Perform these tasks for software, data, and maybe for computer hardware Measure and predict defects; make changes to improve quality

SWEs Should Architect, Design, and Implement SW in a Systems Context

- Security vulnerabilities, countermeasures, patterns, code review, static testing…
- Understand SysE practices and coordinate
- Design for modifiability
- Adapt existing software
- Maintain
- Make defensible decisions (via trade studies or other)
- Ask the right questions

SysEs Should Adapt Systems Engineering Practices to Include Software Engineers

- Identify SW architect roles in early system design and ensure they are done by right people
- Include SW options in trade studies
  - Utility functions of SW contributions to options
- Identify, analyze, manage, and mitigate system-level risks
- Ask the right questions

Ask Questions

SysEs:

- What are inputs and outputs of SW process?
- What major architectural decisions are needed, when? Options?
- How reduce risk of cyberattack?
- What SW risks could escalate?
- What does SW need from me?
- Will this change be easy or hard to implement?
- What decisions should be made first?

SWEs:

- How much do project leaders, managers know about Sys&SW?
- What does SysE need to know from SWE?
- To what level of detail will SysE models be decomposed?
- What groups are you assuming SWEs will interface with?
- What are all the specs SW might need to follow?
- How final are the requirements: when might they change?

Vision: high-performance Sys-SW Interface for CPS

- SysEs and SWEs, working closely together, ensure that the best CPS is designed, built, and maintained.
- A chief SWA and chief SysE(or equivalent titles) coordinate, jointly plan what information they need and can provide.
- Timely trade studies, performed jointly, ensure affordability.
- *SW* architectural concerns are satisfied during *system* architecture development.
- *SWEs/SWAs* remain *up to speed* with a rapidly evolving knowledge base *SysEs* stay knowledgeable about *broad domain and customer*
- Jointly identify and escalate *risks*
- Develop system designs in *modeling tools* that interface seamlessly with SWEs' modeling tools.
- SysEs maintain responsibility for the *non-deterministic and emergent needs* of the system
  SWEs help ensure their deterministic and evolving SW meets those needs

29th Annual INCOSE international symposium

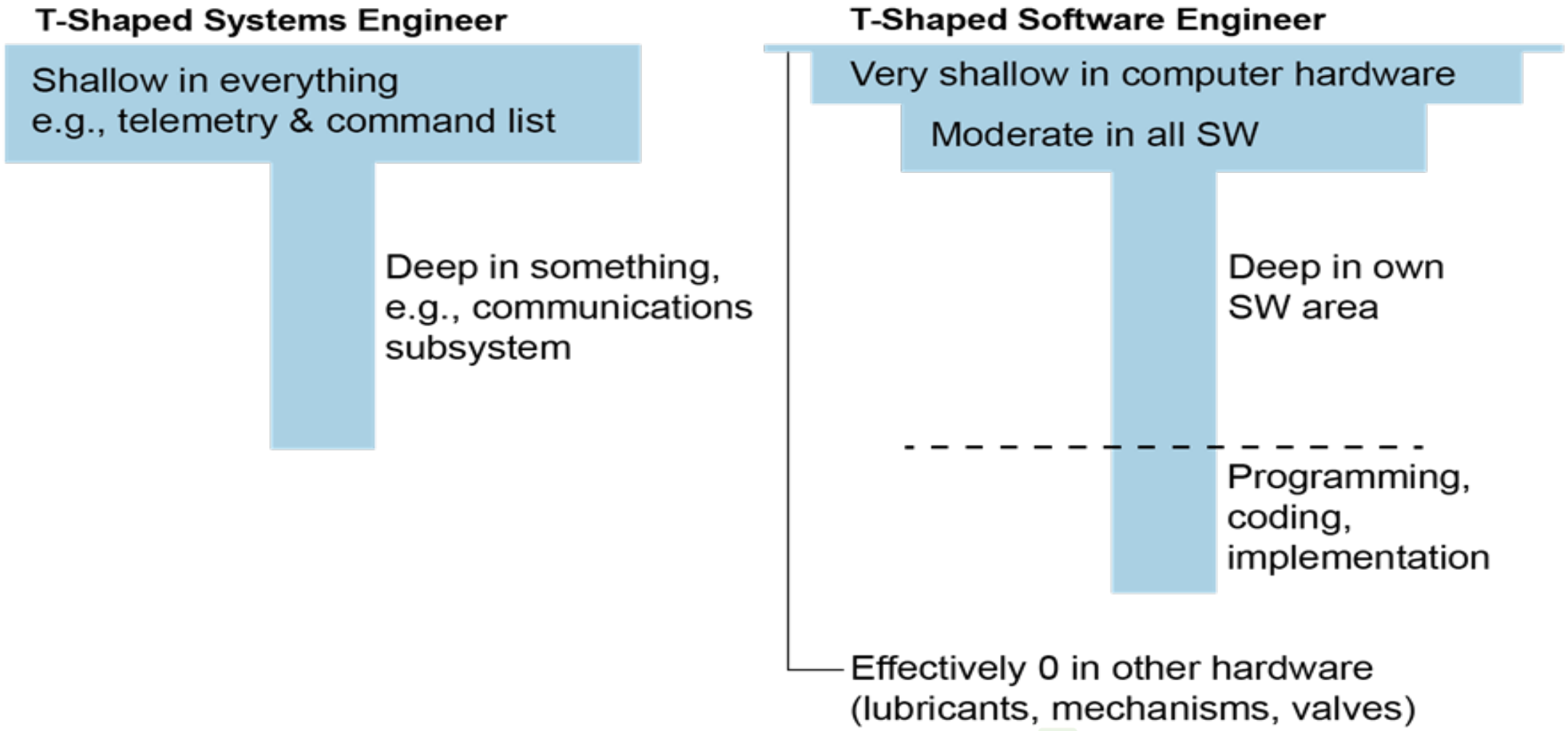
Orlando, FL, USA

July 20 - 25, 2019

Backup

www.incose.org/symp2019

SysE and SWE -1



Engineering

- Definition
- IEEE Code of Ethics: we agree:
  1. to hold paramount the safety, health, and welfare of the public, to strive to comply with ethical design and sustainable development practices, and to disclose promptly factors that might endanger the public or the environment;
  2. to avoid real or perceived conflicts of interest whenever possible, and to disclose them to affected parties when they do exist; (and 8 more)

Systems Engineering

- Post WWII
- Deal with complexity
- Provide top-level view, integration, and balance

- INCOSE Code of Ethics

12 SysE Roles (Sheard 1996)

RO	Requirements owner
SD	System designer
SA	System analyst
VV	V&V engineer
LO	Logistics & ops engineer
G	Glue among subsystems
CI	Customer Interface
TM	Technical manager
IM	Information manager
PE	Process Engineer
CO	Coordinator

Software Engineering

- Term was aspirational, 1968
- Software was:
  - Small programs on specific computers
- Software is now:
  - tiny through huge programs,
  - running in “the cloud,”
  - basis of nearly all new functionality in systems

- Where is discipline, professsionalism, code of ethics?
- How engineer safety, security, maintainability, quality?
- Discontinued professional exam* for SWEs as of 2019

*National Council of Examiners for Engineering and Surveying

Major Problem: Toyota Throttle Software

Toyota Unintended Acceleration and the Big Bowl of "Spaghetti" Code

Posted on Thursday, Nov 7th, 2013 Safety Research & Strategies, Inc.*

Last month, Toyota hastily settled an Unintended Acceleration lawsuit – hours after an Oklahoma jury determined that the automaker acted with "reckless disregard," and delivered a $3 million verdict to the plaintiffs – but before the jury could determine punitive damages.

What did the jury hear that constituted such a gross neglect of Toyota's due care obligations? The testimony of two plaintiff's experts in software design and the design process gives some eye-popping clues. After reviewing Toyota's software engineering process and the source code for the 2005 Toyota Camry, both concluded that the system was defective and dangerous, riddled with bugs and gaps in its failsafes that led to the root cause of the crash….

* http://www.safetyresearch.net/blog/articles/toyota-unintended-acceleration-and-big-bowl-%E2%80%9Cspaghetti%E2%80%9D-code

Major Problem: Toyota Throttle Software

Toyota Unintended Acceleration and the Big Bowl of "Spaghetti" Code

Posted on Thursday, Nov 7th, 2013

…

The accepted, albeit voluntary, industry coding standards were first set by Motor Industry Software Reliability Association (MISRA) in 1995. Accompanying these rules is an industry metric, which equates broken rules with the introduction of a number of software bugs: For every 30 rule violations, you can expect on average three minor bugs and one major bug. Toyota made a critical mistake in declining to follow those standards, he said.

When NASA software engineers evaluated parts of Toyota's source code during their NHTSA contracted review in 2010, they checked 35 of the MISRA-C rules against the parts of the Toyota source to which they had access and found 7,134 violations. Barr checked the source code against MISRA's 2004 edition and found 81,514 violations….

…search.net/blog/articles/toyota-unintended-acceleration-and-big-bowl-%E2%80%9Cspaghetti%E2%80%9D-code

29th Annual INCOSE international symposium

Orlando, FL, USA

July 20 - 25, 2019

www.incose.org/symp2019