



29th Annual **INCOSE**
international symposium

Orlando, FL, USA
July 20 - 25, 2019

Sarah A. Sheard, Ph.D.

Carnegie Mellon University Software Engineering Institute

Complexity in a Systems Engineering Context



What Is Complexity? (1): Definitions

Complexity (n): quality or state of being complex^(1,2)

Complex (adj): composite; hard to separate, analyze, or solve⁽¹⁾

Complex (adj): consisting of interconnected or interwoven parts; intricate in structure, complicated⁽²⁾

Objective: Many pieces, nonlinear, self-organizing, decentralized...

Subjective: Difficult to analyze, difficult to understand, risky...

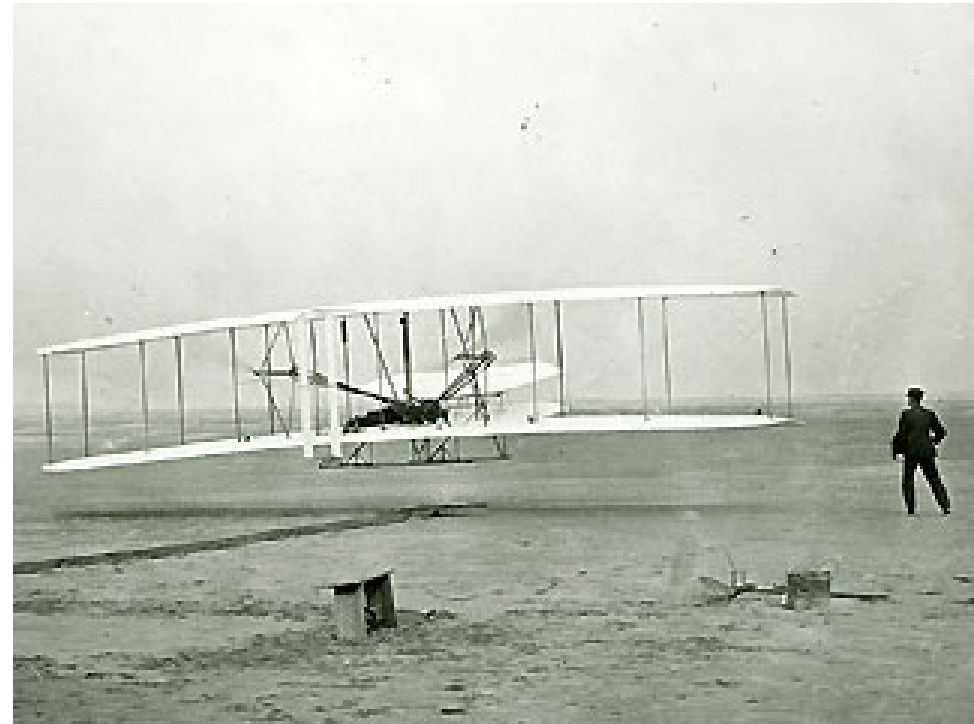
(1) Merriam-Webster Collegiate Dictionary

(2) American Heritage Dictionary



What Is Complexity? (2): Caution

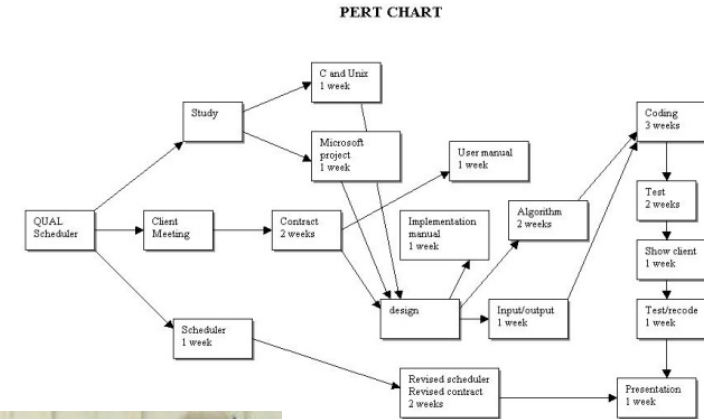
***Complexity is not a thing;
... it is a characteristic of things***





What Is Complexity? (2): Entities*

- The **system** being built
- The **project** building it
- The **environment** it will affect
 - Technical
 - Socio-political



Taxpayers

- **Cognitive aspects** (confusion, frustration, difficulty)

*(Sheard 2012)



Systems Engineering and Complexity

- SysE for complexity management is not new
 - Hall (1962): Purpose of SysE is to manage complexity
 - Techniques mostly not new: Complex adaptive systems, systems of systems
 - Volume, variety and dynamics of information, stakeholders, uses tend to be new



How Should Systems Engineers Use Complexity?

- *Want one chart*: where do various kinds of complexity relate to the systems engineering domain
- e.g., 33 definitions from Young, Farr, and Valerdi* (created for measurement)

*Young, L. Z., Farr, J. V., & Valerdi, R. 2010. "The role of complexities in systems engineering cost estimating processes." Paper presented at the conference on systems engineering research, Hoboken NJ (US), 17-19 March.

33 Definitions (Young, Farr, and Valerdi 2010)



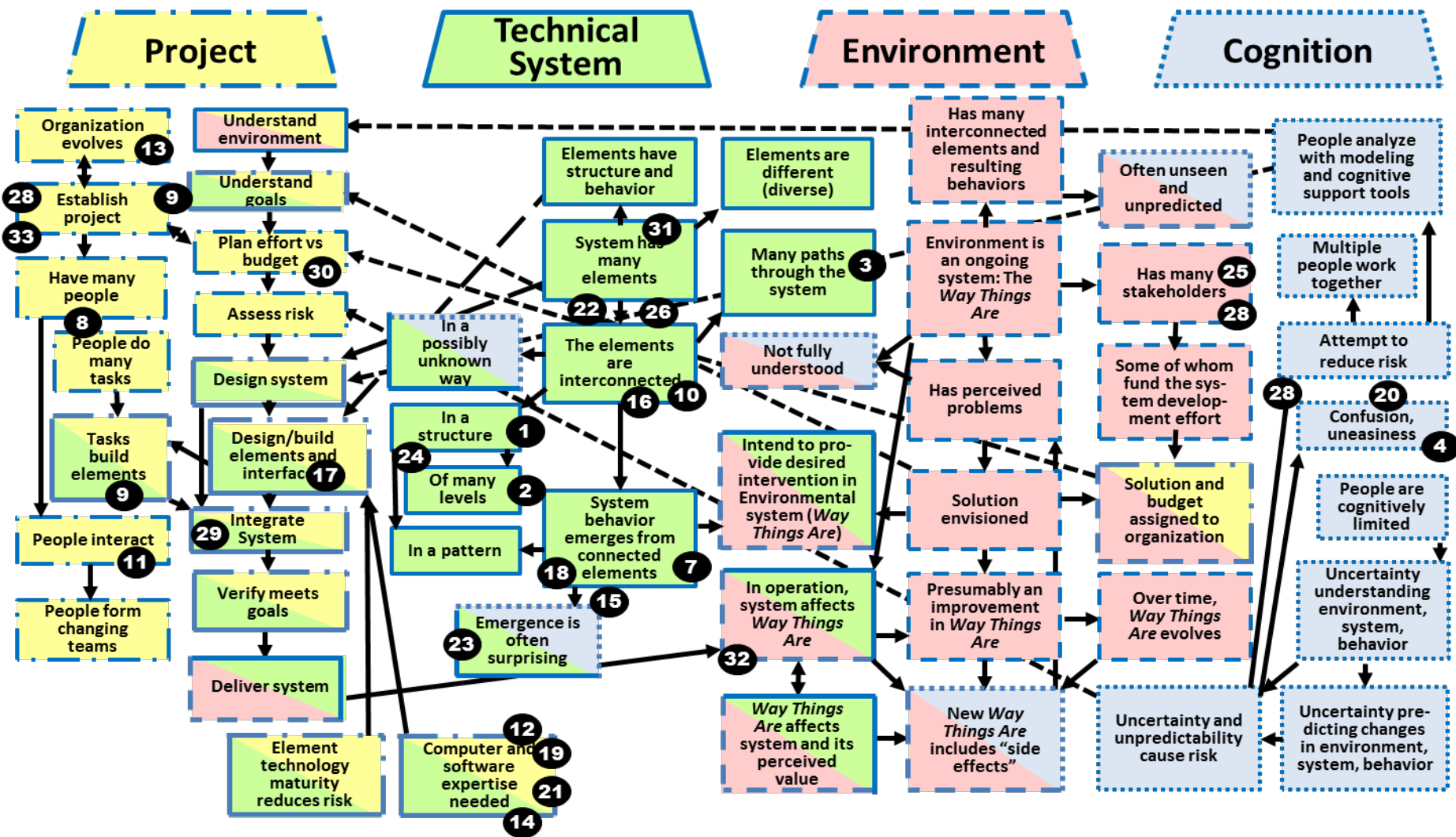
Type of Complexity

- 1 Hierarchical/Structural (# levels)
- 2 Configuration Complexity
- 3 Complicatedness/ Functional Complexity
- 4 Subjective Complexity
- 5 Statistical Complexity
- 6 Algorithmic/Deterministic Complexity
- 7 Aggregate Complexity (interrelationships)
- 8 Project Complexity (organizational and technological)
- 9 Project complexity (assembly, system, array)
- 10 Product Complexity (physical)
- 11 Structural Organizational Complexity
- 12 Structural IT Complexity
- 13 Dynamic Organizational Complexity
- 14 Dynamic IT Complexity
- 15 Inter-Component Complexity (can grow exponentially)
- 16 Interface Complexity (by component)
- 17 Implementation Complexity (e.g. code)

Type of Complexity

- 18 System-level Complexity (emergent)
- 19 Structural Complexity (design and structure, persistent)
- 20 Conceptual Complexity (psychological)
- 21 Computational Complexity (algorithms)
- 22 Structural/Combinatorial Complexity
- 23 Behavioral Complexity (unpredictability)
- 24 Nested Complexity (technical/socio-technical)
- 25 Evaluative Complexity (multiple stakeholder viewpoints)
- 26 Static Complexity
- 27 Dynamic Complexity
- 28 Social-Political Complexity
- 29 Technical Complexity (Systems Integration based)
- 30 Programmatic Complexity (Systems Integration based)
- 31 Configuration Complexity (Systems Integration based)
- 32 Operational Complexity (Systems Integration based)
- 33 Organizational Complexity (Systems Integration based)

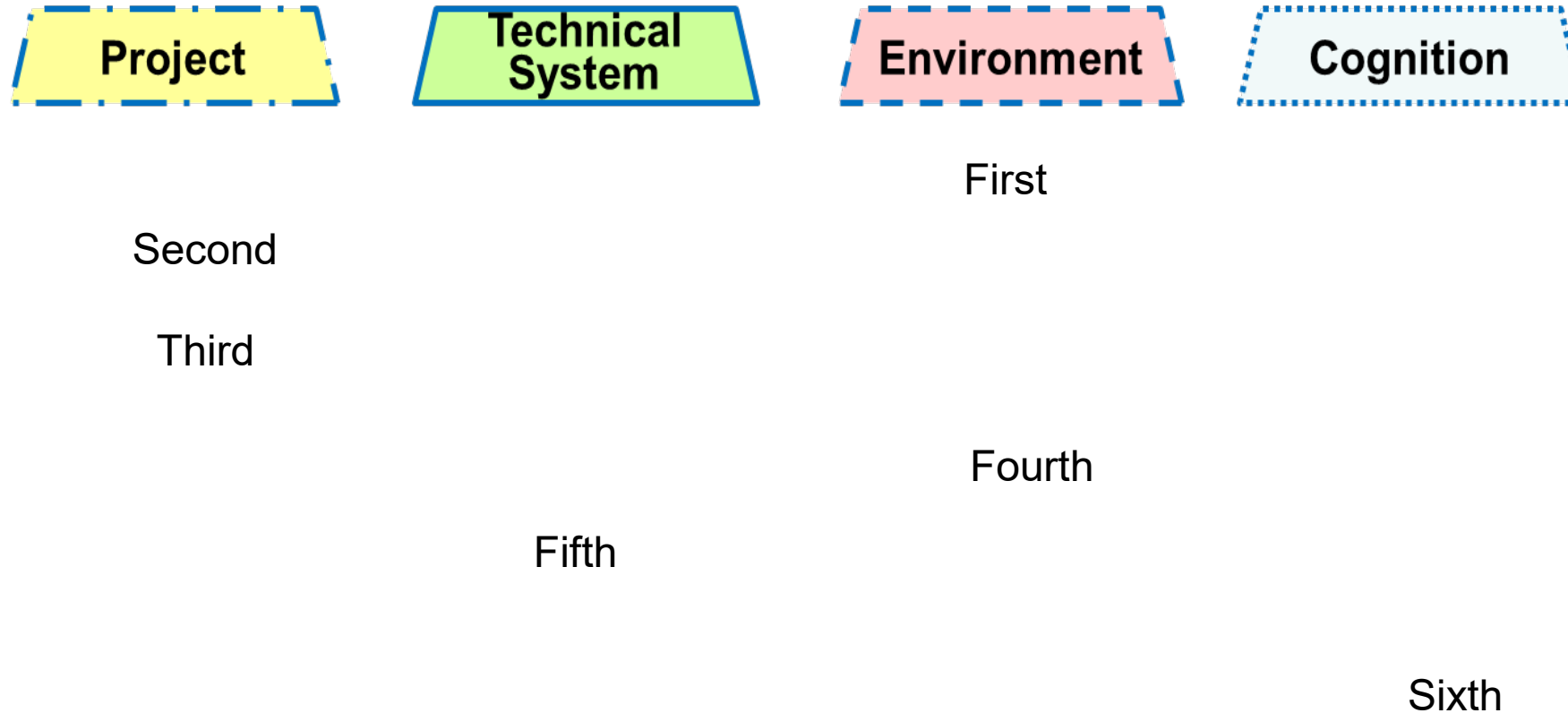
One Chart (I will walk through this)



- All these things change over time: 13, 14, 27, 31
- All these things have information (data): 5, 6

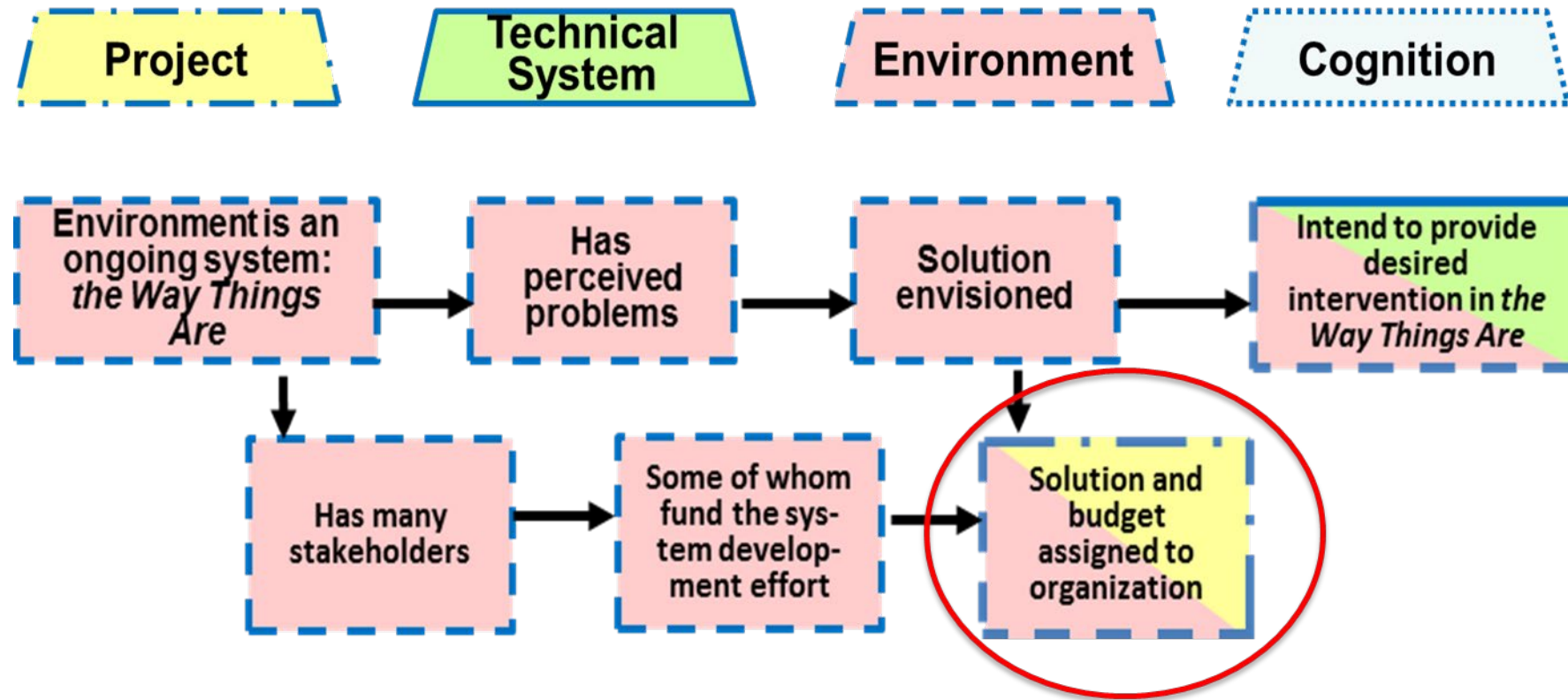


Complexity as an Adjective, not a Noun



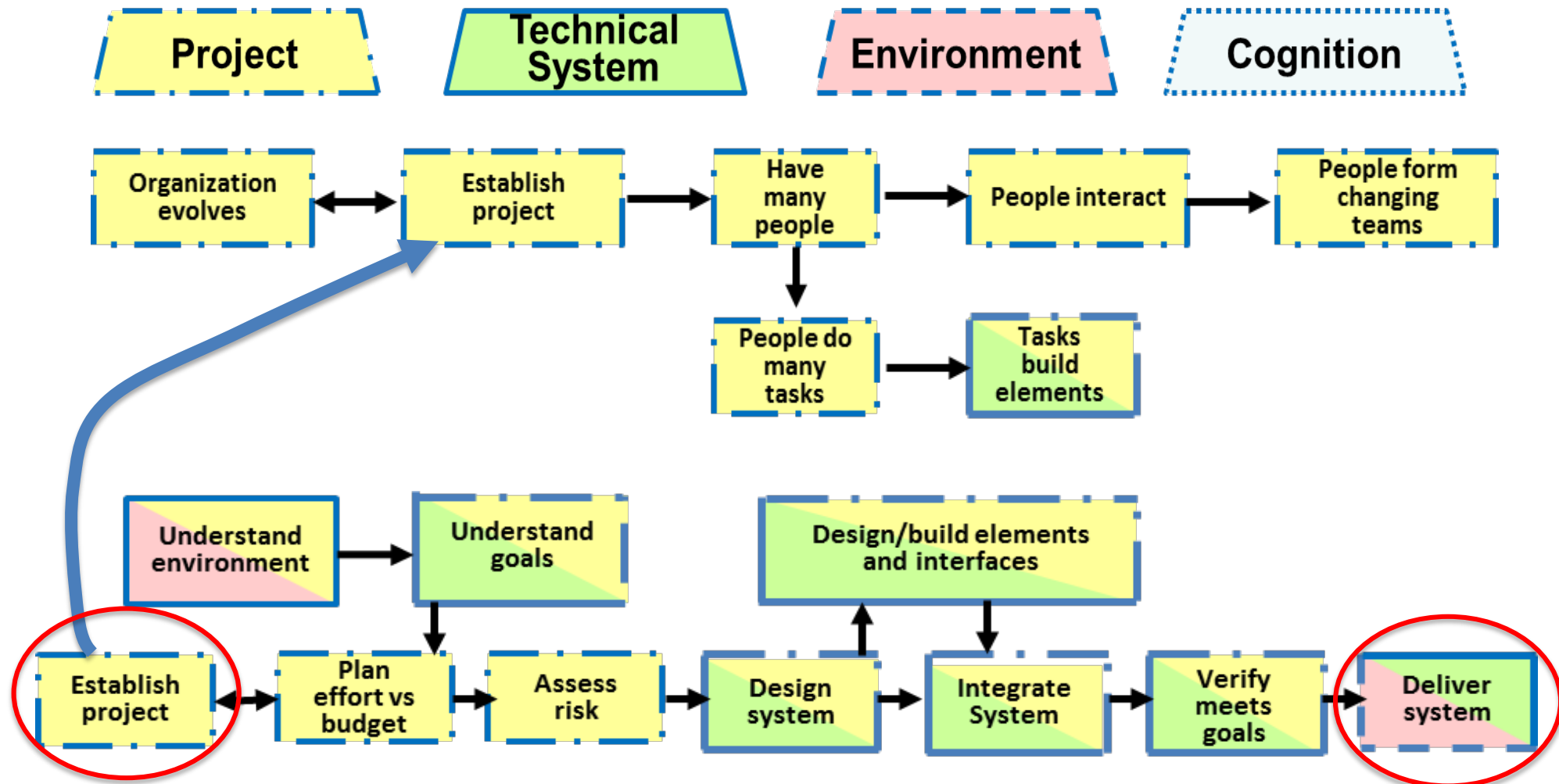


Environment Elements (1)



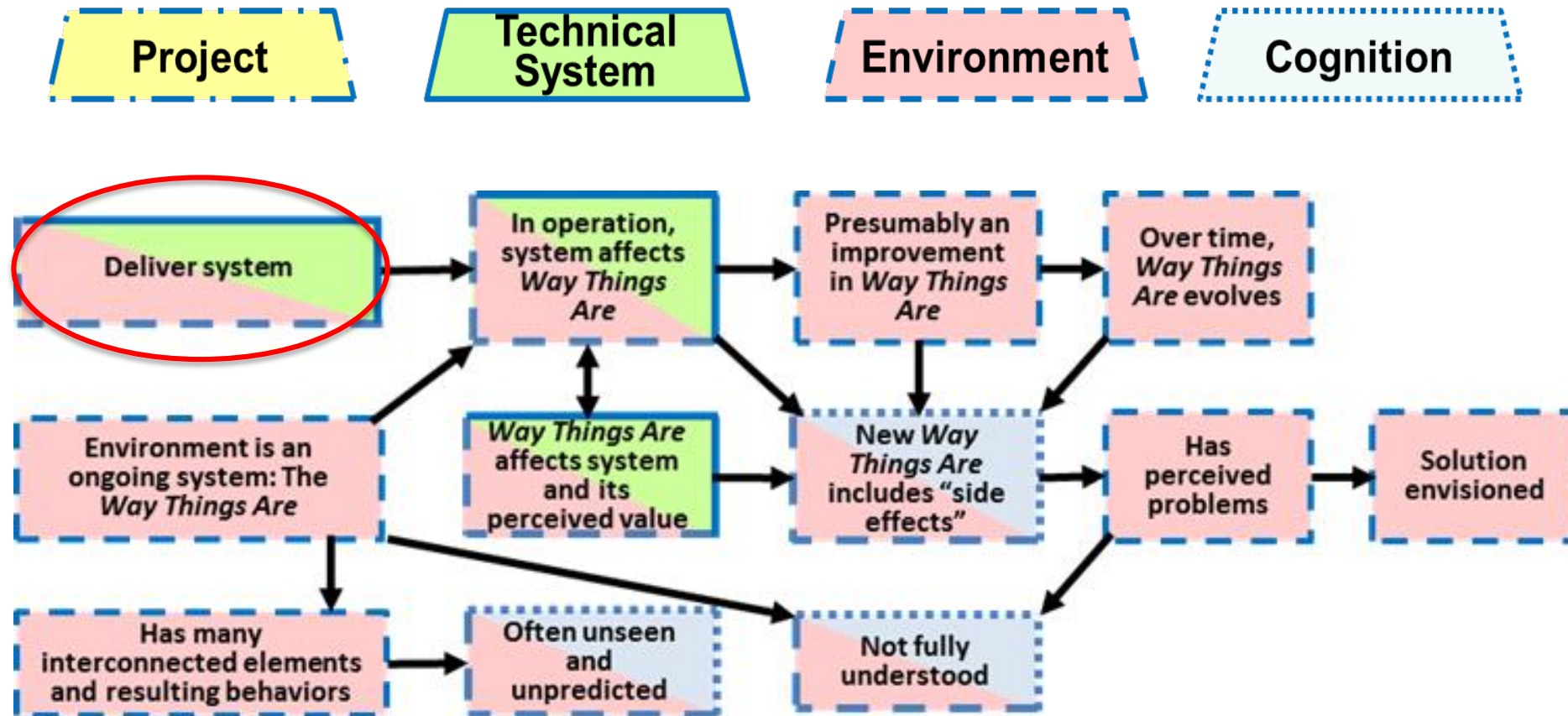


Project Elements(1) and (2)



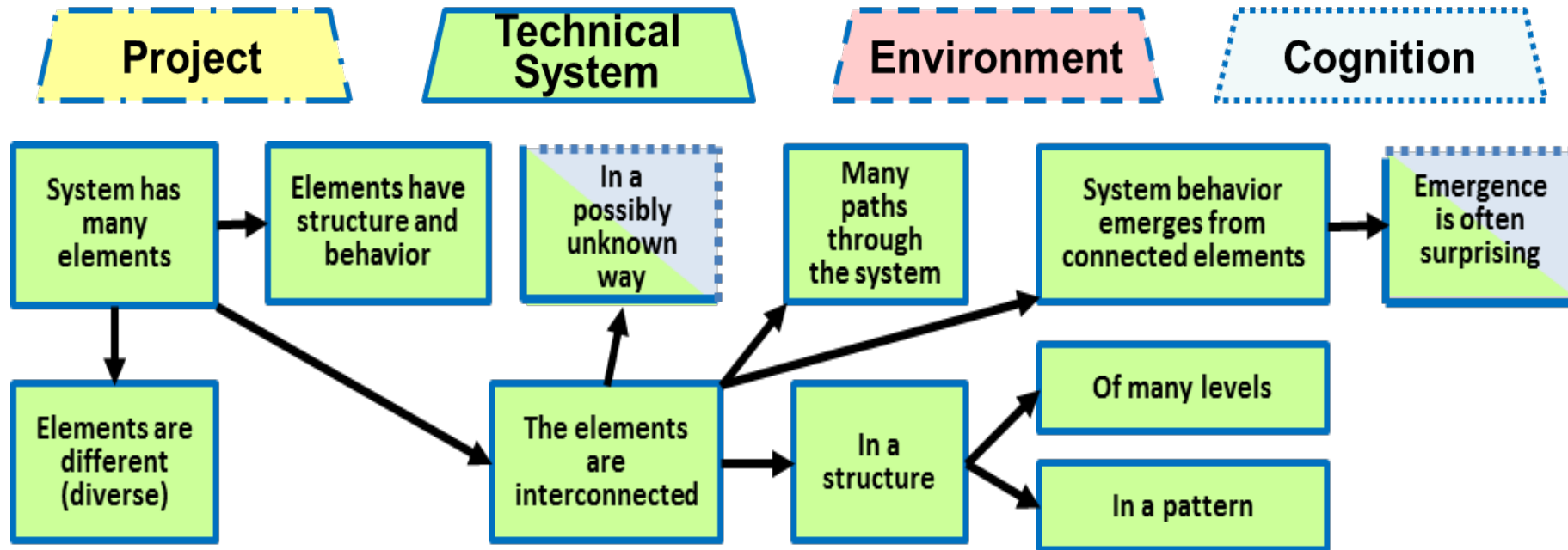


Environment Elements (2)



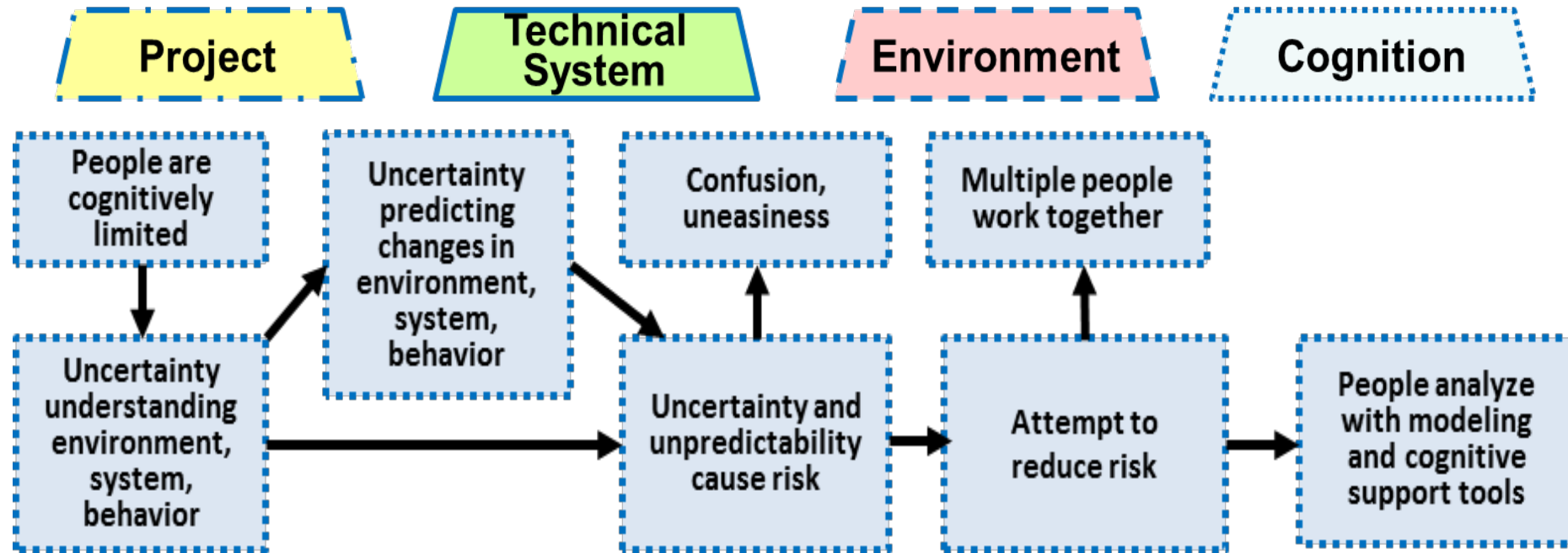


System Activities



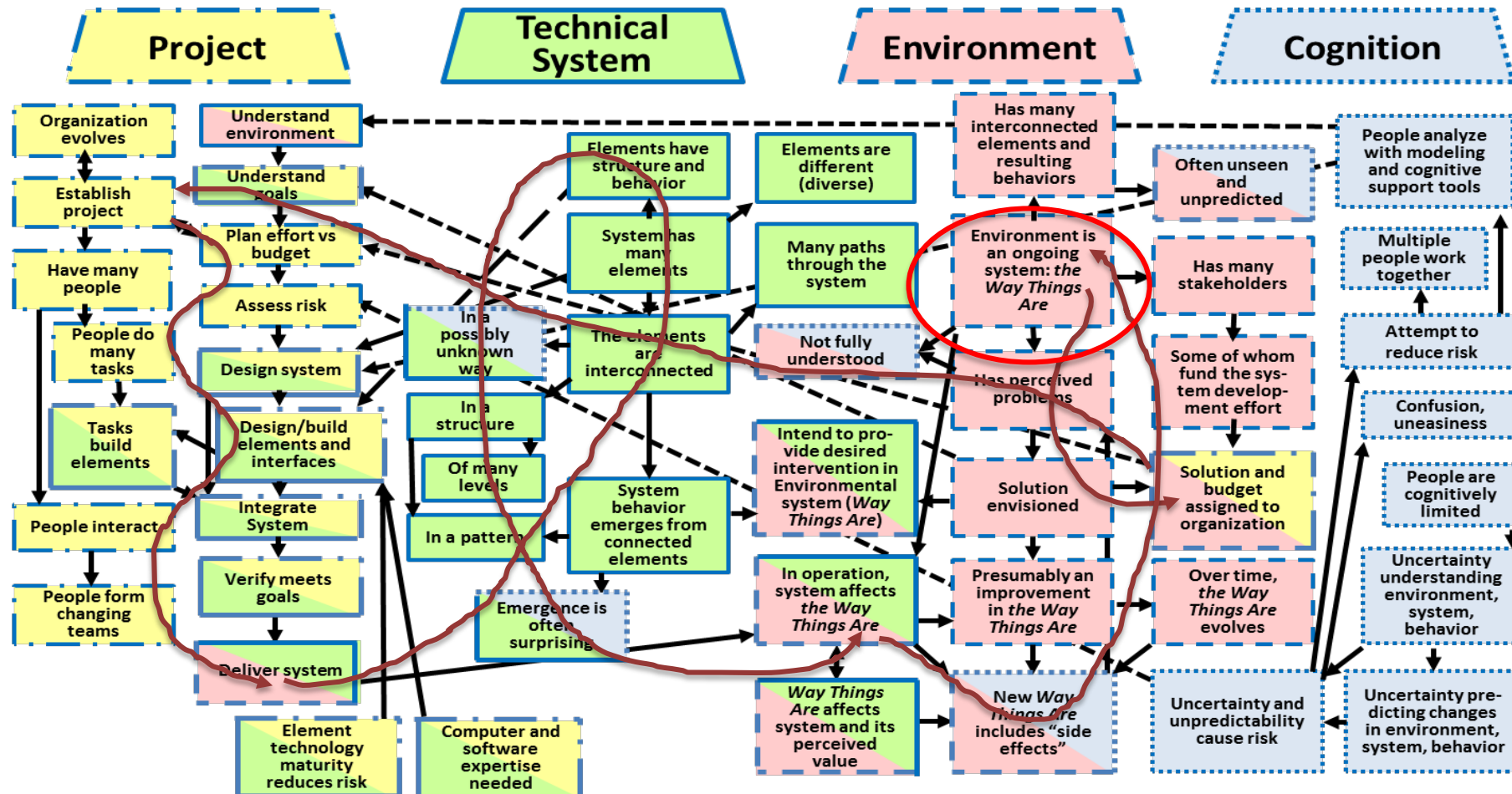


Cognitive Activities





Recap: A System Lifecycle?



33 Definitions (Young, Farr, and Valerdi 2010)

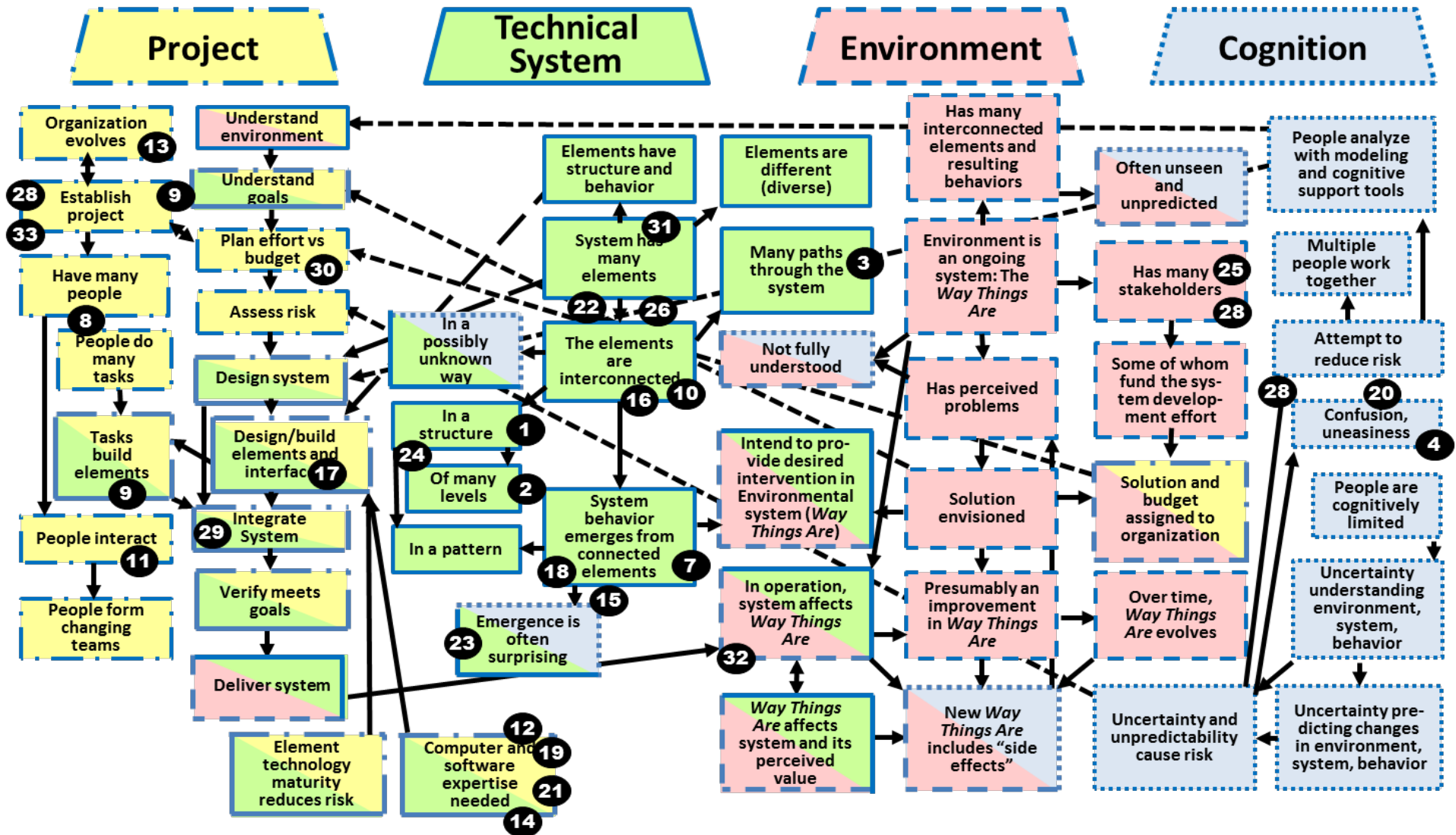


Type of Complexity

- 1 Hierarchical/Structural (# levels)
- 2 Configuration Complexity
- 3 Complicatedness/ Functional Complexity
- 4 Subjective Complexity
- 5 Statistical Complexity
- 6 Algorithmic/Deterministic Complexity
- 7 Aggregate Complexity (interrelationships)
- 8 Project Complexity (organizational and technological)
- 9 Project complexity (assembly, system, array)
- 10 Product Complexity (physical)
- 11 Structural Organizational Complexity
- 12 Structural IT Complexity
- 13 Dynamic Organizational Complexity
- 14 Dynamic IT Complexity
- 15 Inter-Component Complexity (can grow exponentially)
- 16 Interface Complexity (by component)
- 17 Implementation Complexity (e.g. code)

Type of Complexity

- 18 System-level Complexity (emergent)
- 19 Structural Complexity (design and structure, persistent)
- 20 Conceptual Complexity (psychological)
- 21 Computational Complexity (algorithms)
- 22 Structural/Combinatorial Complexity
- 23 Behavioral Complexity (unpredictability)
- 24 Nested Complexity (technical/socio-technical)
- 25 Evaluative Complexity (multiple stakeholder viewpoints)
- 26 Static Complexity
- 27 Dynamic Complexity
- 28 Social-Political Complexity
- 29 Technical Complexity (Systems Integration based)
- 30 Programmatic Complexity (Systems Integration based)
- 31 Configuration Complexity (Systems Integration based)
- 32 Operational Complexity (Systems Integration based)
- 33 Organizational Complexity (Systems Integration based)



All these things change over time 13 14 27 31

All these things have information (data) 5 6



Takeaways

- Complexity is an adjective
- More than just the system, or the project, is important
- Sometimes you can't control but you can influence; sometimes just work around
- All ways to measure complexity are incomplete



Using Knowledge of Complexity

- Identify relative complexity and relative risk
 - This is more complex than we thought
 - This module is more complex than that module
 - This social complexity is riskier than the technical risk
- Identify specific risks
 - Risk of stakeholder conflicts changing our requirements
 - Risk of conflicting requirements buried in referenced documents
- Identify kinds of complexity and address as risks
 - Interface definition is a risk, should be attended to early
- Consider using currently collected metrics
 - Requirements volatility, %Change actions approved, Time to repair defects...



Summary

- Complexity refers to many entities and has both technical and cognitive aspects
- A system 'lifecycle' that includes the environment and the project can address, *or at least recognize*, most types of complexity
- Not all views on one chart

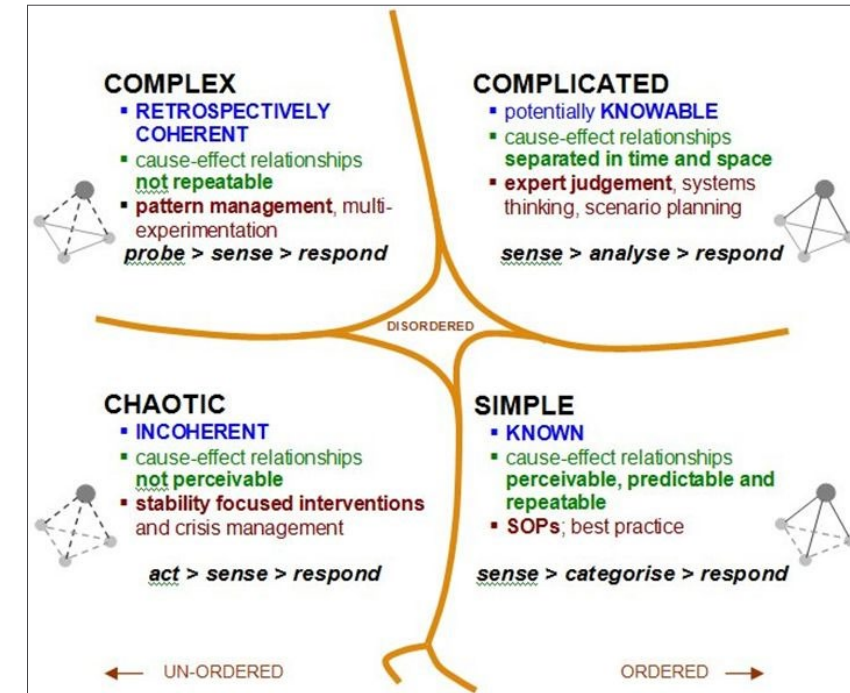


Backups

Why I'm Not Talking Complex vs. Complicated

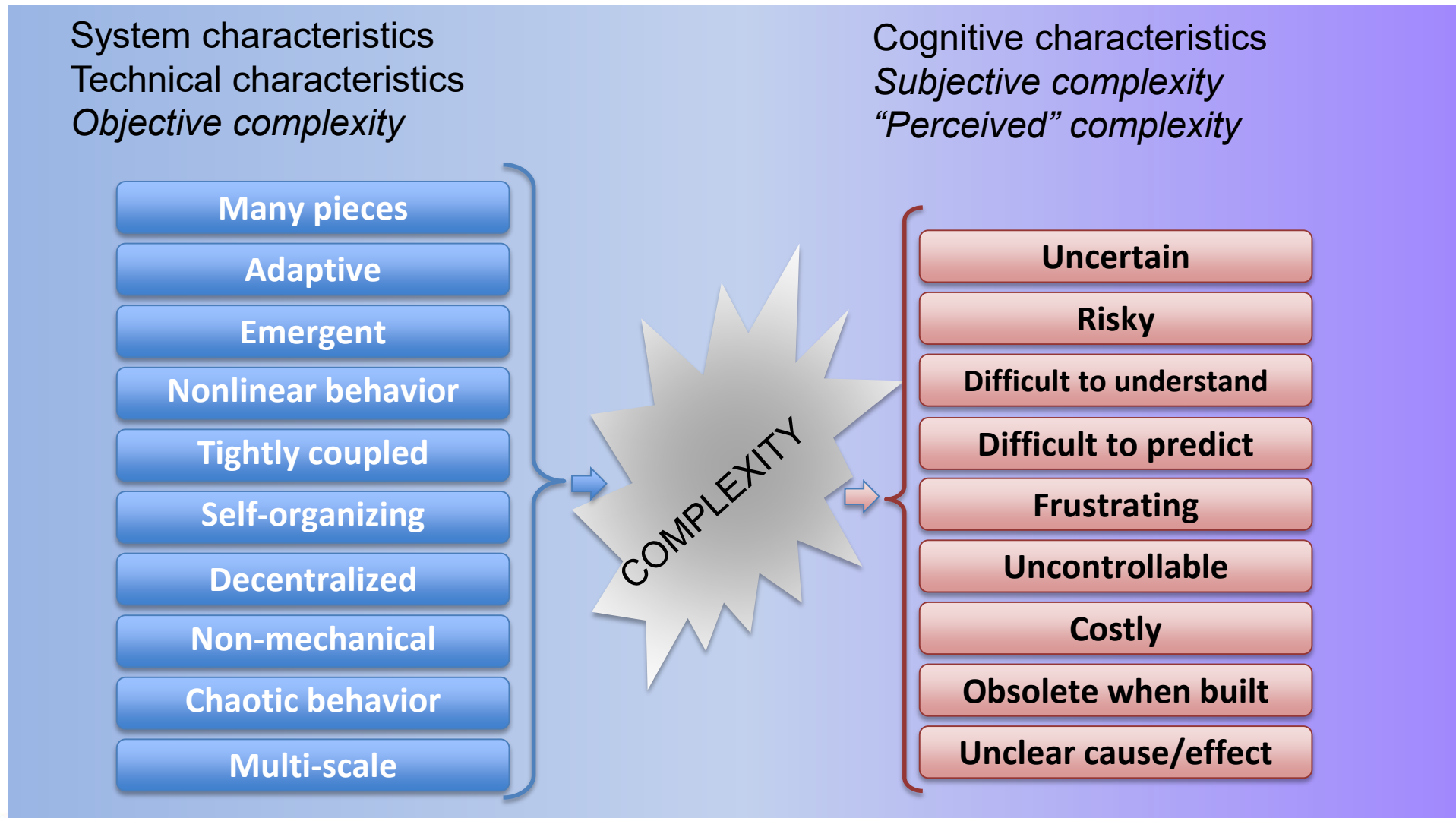


- “Complicated” means many things
 - “Can use same practices, only more of them” = MITRE (Stevens)
 - Realm of systems analysis (Cynefin framework, by Kurtz and Snowden)
 - Overloaded and sometimes reversed:
 - “Complexity is intrinsic, complicated is because of external influences”
 - “Complexity does not evoke difficulty; complicated refers to a high level of difficulty”
 - Definitions change with time: Yesterday’s complex is today’s complicated, and maybe neither in the future
 - Seems to be too much shorthand. “Complicated” means “what I’m not talking about” and “Complex” means “what I am talking about.”
- I consider “Complex” to be a spectrum





What Is Complexity? (1) Objective–Subjective

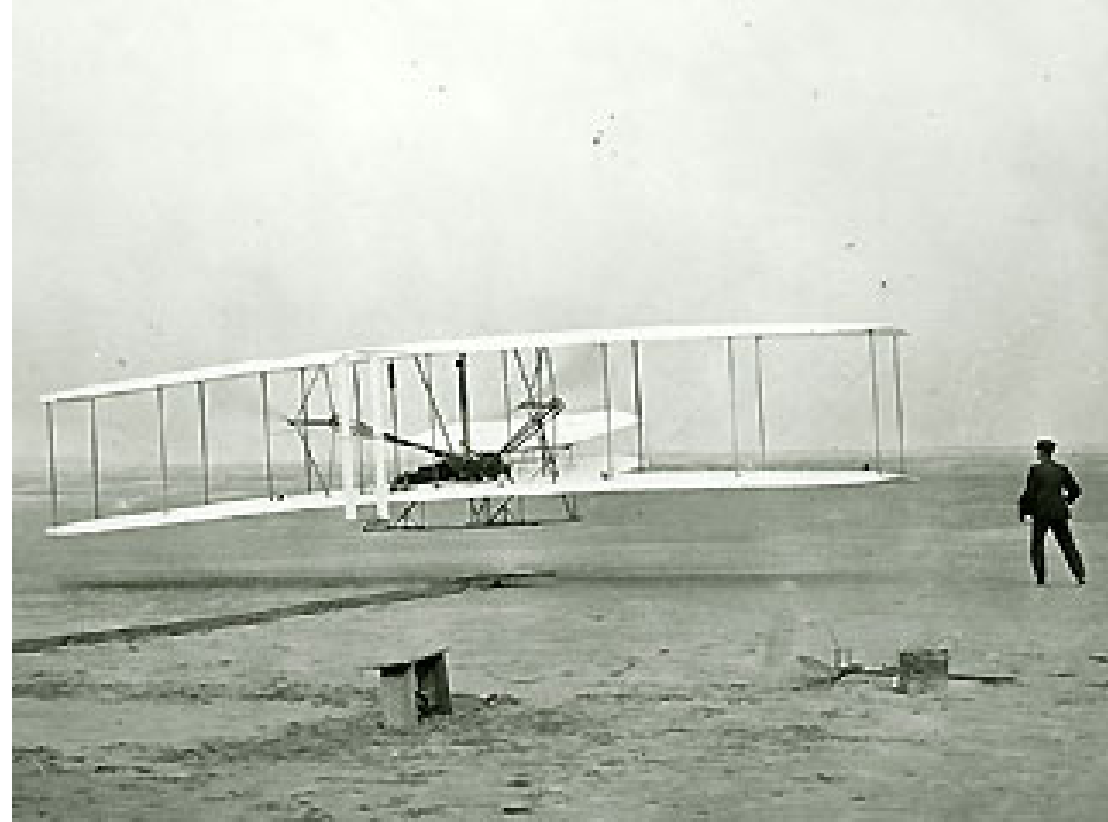




What Is Complexity? (2): Definitions

- Complexity, defined *subjectively*, relentlessly decreases
- Complexity, however defined *objectively*, relentlessly increases

Yet we manage it





References

Slide 3 (1) Merriam-Webster Collegiate Dictionary

Slide 3 (2) American Heritage Dictionary

Sheard 2012 (Dissertation)

Arthur Hall 1962 (paper)

Young, Farr, and Valerdi 2010



Contact Information

Sarah Sheard
Software Engineering Institute
sarah.sheard@gmail.com
(703) 994 7284



29th Annual **INCOSE**
international symposium

Orlando, FL, USA

July 20 - 25, 2019

www.incose.org/symp2019



Copyright 2019 Carnegie Mellon University. All Rights Reserved.

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.

NO WARRANTY. THIS CARNEGIE MELLON UNIVERSITY AND SOFTWARE ENGINEERING INSTITUTE MATERIAL IS FURNISHED ON AN "AS-IS" BASIS. CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT.

[DISTRIBUTION STATEMENT A] This material has been approved for public release and unlimited distribution. Please see Copyright notice for non-US Government use and distribution.

This material may be reproduced in its entirety, without modification, and freely distributed in written or electronic form without requesting formal permission. Permission is required for any other use. Requests for permission should be directed to the Software Engineering Institute at permission@sei.cmu.edu.

DM19-0474