Complexity, Systems, and Software

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

Sarah A. Sheard, Ph.D.
2014
# Complexity, Systems, and Software

**Author:** Sheard /Sarah

**Performing Organization:**
Software Engineering Institute Carnegie Mellon University
Pittsburgh, PA 15213

**Distribution/Availability Statement:**
Approved for public release, distribution unlimited.

**Notes:**
The original document contains color images.

**Security Classification:**
- Report: unclassified
- Abstract: unclassified
- This Page: unclassified
- Limitation of ABSTRACT: SAR
- Number of PAGES: 26
Agenda

What is complexity?
Complexity and project outcomes
Complexity of systems and software
Changing nature of systems and software
What Is Complexity?

(1) Objective–Subjective
(2) Definitions
(3) Entities
(4) Types
What Is Complexity?

(1) Objective–Subjective

System characteristics
Technical characteristics

*Objective complexity*

- Many pieces
- Adaptive
- Emergent
- Nonlinear behavior
- Tightly coupled
- Self-organizing
- Decentralized
- Non-mechanical
- Chaotic behavior
- Multi-scale

Cognitive characteristics

*Subjective complexity*

- Uncertain
- Risky
- Difficult to understand
- Difficult to predict
- Frustrating
- Uncontrollable
- Costly
- Obsolete when built
- Unclear cause/effect
Proposition: However you define complexity, your definition is incomplete

Don’t call anything “complexity”

At least call it “X” complexity

Proposition: Engineering seeks complexity management; complexity reduction is one way of doing that

SysE for complexity reduction is not new

- Hall (1962): purpose of SysE is to manage complexity
- Techniques mostly not new: Complex adaptive systems, systems of systems
What Is Complexity? (2): Definitions

Complexity, defined \textit{subjectively}, relentlessly decreases

Complexity, however defined \textit{objectively}, relentlessly increases

Yet we manage it

\textit{Proposition: Complexity is not a thing} 
\textit{... it is a characteristic of things}
What Is Complexity? (3): Entities

The **system** being built

The **project** building it

The **environment** it will affect

- Technical
- Socio-political

Cognitive aspects (confusion, frustration, difficulty)
What Is Complexity? (4): Types*

Structural
• Size (# parts, stakeholders, elements, LOC)
• Connectivity (# or density, types, strength of connection)
• Inhomogeneity (diversity, architecture, loops…)

Dynamic
• Short-term (e.g., behavioral nonlinearity)
• Long-term (evolution, transition to new states)

Socio-political
• Organizational maturity, stakeholder conflict, global context…

*(Sheard 2012)
**Strike a Balance**

*Proposition: The point of engineering is control*

*Proposition: Complexity has no good side*

- Study it to recognize it, to manage it, to reduce it

But: being overly simple is also wrong

Ashby’s Law of Requisite Variety: A control system must have at least as many degrees of freedom as the disturbances it needs to counteract

- Technical system shouldn’t be too simple (Allocating all complexity to operator)
- Technical system shouldn’t be too complex (Hidden issues; dumbs down operator)
39 Complexity Questions (Sheard 2012)*

# Subsystems
# Easy, nominal, difficult requirements
Technology maturity
Architecture precedence
Schedule margin
Staff skills

# Sponsors
Stakeholder conflict
Stakeholder relationships
Cognitive fog

Other questions
• Project outcomes (cost, schedule, performance, subjective assessment of outcome, produce a product)
• Project start/end dates
• Project size (cost)
• Management methods (plan, risk, agile, lean, set-based)
• Respondent role and confidence

75 programs: Did complexity correlate to cost, schedule, or performance problems?

*Sheard, Sarah A. Assessing the impact of complexity attributes on system development project outcomes. Dissertation, Stevens Institute of Technology, School of Systems and Enterprises, May 2012.
## Results: Top 3 Correlating Questions

<table>
<thead>
<tr>
<th>Complexity Variable</th>
<th>Outcome Variable</th>
<th>Cost Overrun</th>
<th>Schedule Delay</th>
<th>Performance Shortfall</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Q16d—Requirements Difficult</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Low (Under 100) group mean</td>
<td></td>
<td>3.37</td>
<td>3.30</td>
<td>2.26</td>
</tr>
<tr>
<td>High (Over 100) group mean</td>
<td></td>
<td>5.00</td>
<td>4.64</td>
<td>3.60</td>
</tr>
<tr>
<td>p-value</td>
<td></td>
<td>0.00027</td>
<td>0.00165</td>
<td>0.00163</td>
</tr>
<tr>
<td>Significance</td>
<td></td>
<td><strong>p&lt;0.001</strong></td>
<td><strong>p&lt;0.05</strong></td>
<td><strong>p&lt;0.05</strong></td>
</tr>
<tr>
<td><strong>Q32—Cognitive Fog</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Low (D-SD) group mean</td>
<td></td>
<td>3.03</td>
<td>2.97</td>
<td>2.00</td>
</tr>
<tr>
<td>High (A-SA) group mean</td>
<td></td>
<td>3.89</td>
<td>4.11</td>
<td>3.53</td>
</tr>
<tr>
<td>p-value</td>
<td></td>
<td>0.0395</td>
<td>0.0120</td>
<td>0.00074</td>
</tr>
<tr>
<td>Significance</td>
<td></td>
<td><strong>p&lt;0.05</strong></td>
<td><strong>p&lt;0.05</strong></td>
<td><strong>p&lt;0.001</strong></td>
</tr>
<tr>
<td><strong>Q38f—Stakeholder Relationships</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Low (Stable) group mean</td>
<td></td>
<td>3.30</td>
<td>3.11</td>
<td>2.15</td>
</tr>
<tr>
<td>High (Resistance) group mean</td>
<td></td>
<td>4.50</td>
<td>4.19</td>
<td>3.27</td>
</tr>
<tr>
<td>p-value</td>
<td></td>
<td>0.0209</td>
<td>0.0243</td>
<td>0.0245</td>
</tr>
<tr>
<td>Significance</td>
<td></td>
<td><strong>p&lt;0.05</strong></td>
<td><strong>p&lt;0.05</strong></td>
<td><strong>p&lt;0.05</strong></td>
</tr>
</tbody>
</table>
Complexity of Systems and Software

Software: McCabe (cyclomatic) complexity: decisions in a code function
  - Paths ~ edges and nodes
  - Used to estimate defects & reliability

Systems: No complexity metric available

Proposition: Measurement is inherently simplification.
Measurement of complexity is like describing Red by means of Green variables

Use knowledge of complexity:
  - Identify relative complexity and relative risk
  - Identify specific risks
  - Identify kinds of complexity and address as risks
  - Probably tie to currently collected metrics, e.g., requirements volatility
Dealing with Complexity

Determine what kind
Apply systems engineering principles and practices
Identify any special complexity as a risk
Study how to other fields manage that risk
  • Bring in experts

Today’s “New” complexity:
Emphasis shift from “whole system” to software
  • What is it?
  • How should systems and software engineers manage it?
Changing Nature of Systems and Software

- **1970s**
- **1980s**
- **1990s**
- **2000s**
- **2010s**
- **2020s**

**Red = SW**
**Blue = System**

- USAF NW1
- Army NW3
- Health NW
- SAP
Conclusion

Complexity means many different things
• Countable, technical complexity vs. difficulty

Systems and software are getting ever more complex
• Complexity measures are inadequate
• Systems engineering has always been about managing complexity
• Some program characteristics predict cost & schedule problems; are they true “complexity”? 

Tom Lehrer’s First Law of Thermodynamics applies
• “You can’t win, the best you can do is break even”
Contact Information

Sarah Sheard, Ph.D.
Senior Engineer
Software Solutions Division
Telephone: +1 412-268-7612
Email: sheard@sei.cmu.edu

Web
www.sei.cmu.edu
www.sei.cmu.edu/contact.cfm

U.S. Mail
Software Engineering Institute
Customer Relations
4500 Fifth Avenue
Pittsburgh, PA 15213-2612
USA

Customer Relations
Email: info@sei.cmu.edu
SEI Phone: +1 412-268-5800
SEI Fax: +1 412-268-6257
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
Changing Nature of Systems and Software: Needed Skills

T-Shaped Systems Engineer

Shallow in everything
- e.g., Telemetry & Command list

Deep in something,
- e.g., communications subsystem

Proposition: Software engineering = systems engineering of software system plus implementation

T-Shaped Software Engineer

Very shallow in computer hardware

Moderate in all SW

Deep in own SW area

Programming, Coding, Implementation

Effectively 0 in other hardware (lubricants, mechanisms, valves)
### Complexity Questions

<table>
<thead>
<tr>
<th>#</th>
<th>Complex Variable, Question</th>
<th>Answer Choices</th>
</tr>
</thead>
<tbody>
<tr>
<td>16d</td>
<td>Requirements, Difficult</td>
<td>1-10 10-100 100-1000 1000-10,000 Over 10,000</td>
</tr>
<tr>
<td></td>
<td>Approximately how many system-level requirements did the project have initially? Difficult</td>
<td></td>
</tr>
<tr>
<td></td>
<td>requirements are considered difficult to implement or engineer, are hard to trace to</td>
<td></td>
</tr>
<tr>
<td></td>
<td>source, and have a high degree of overlap with other requirements. How many system</td>
<td></td>
</tr>
<tr>
<td></td>
<td>requirements were there that were Difficult?</td>
<td></td>
</tr>
<tr>
<td>32</td>
<td>Cognitive Fog</td>
<td>Strongly Agree Agree</td>
</tr>
<tr>
<td></td>
<td>‘The project frequently found itself in a fog of conflicting data and cognitive overload.’</td>
<td>Neutral Disagree Strongly Disagree</td>
</tr>
<tr>
<td></td>
<td>Do you agree with this statement?</td>
<td></td>
</tr>
<tr>
<td>38f</td>
<td>Stakeholder Relationships</td>
<td>Relationship stable</td>
</tr>
<tr>
<td></td>
<td>“Where did your project fit, on a scale of Traditional, Transitional, or Messy Frontier,</td>
<td>New Relationships</td>
</tr>
<tr>
<td></td>
<td>in the following eight attributes?” 38f. “Stakeholder relationships: 1: Relationships</td>
<td>Resistance to Changing</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Outcome Questions

<table>
<thead>
<tr>
<th>Outcome Variables</th>
<th>Outcome Questions</th>
<th>Under cost</th>
<th>At cost, +/- 5%</th>
<th>5-20% over plan</th>
<th>20-50% over</th>
<th>50-100% over</th>
<th>More than 100% over plan</th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
<td><strong>Cost Overrun</strong></td>
<td>Under cost</td>
<td>At cost, +/- 5%</td>
<td>5-20% over plan</td>
<td>20-50% over</td>
<td>50-100% over</td>
<td>More than 100% over plan</td>
</tr>
<tr>
<td></td>
<td>At the point of finishing, how much did the project cost, compared to the initially predicted cost for delivery?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td><strong>Schedule Delay</strong></td>
<td>Ahead of schedule</td>
<td>On time within 5%</td>
<td>5-20% late</td>
<td>20-50% late</td>
<td>50-100% late</td>
<td>More than 100% late</td>
</tr>
<tr>
<td></td>
<td>At the point of finishing, how long had the project taken, compared to the initially scheduled development time?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td><strong>Performance Shortfall</strong></td>
<td>Higher than specified</td>
<td>Same as specified, within 5%</td>
<td>Low by 5-20% (fewer features or waived requirements)</td>
<td>Low by 20-50%</td>
<td>Low by more than 50%, or project was cancelled</td>
<td></td>
</tr>
<tr>
<td></td>
<td>At the point of finishing, how was the project performance, compared to the initially specified performance? (Please consider the average performance of <em>mission critical</em> features, and add any qualifiers in Notes.)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
One Plausible Causal Chain

1. Changing Stakeholder Relationships & Resistance
   - Resistance
   - Changing Relationships

2. New Stakeholder Personnel
3. Power Struggles
4. Late decisions
5. Performance Shortfall
6. Difficult Requirements
   - Hard to trace to source
   - Hard to Implement
   - High Overlap
7. Stakeholder Clarification
8. Political Arguments
9. Requirements changes
10. Instability and Conflicting Data
11. Imperfect solutions
12. Schedule Delay
13. Rework
14. More Tests, More Data
15. Expensive solutions
16. Usable Inexpensive solutions
17. Cognitive Fog
18. Wrong decisions
19. Schedule Delay

Complexity, Systems, and Software
Sarah Sheard             August 14, 2014
© 2014 Carnegie Mellon University
Figure 29. Congruence

*Significant, p<0.05; **Significant, p<0.001. Green: variable complexity rises together; Red: opposite; Yellow: neither. Blue=outcome variables; Beige=hypothesis variables.
What Is Complexity?

Webster: “the quality or state of being complex”

• Complex: Composite; hard to separate, analyze, or solve; concerning complex numbers

DARPA: Parts count + SLOC

Algorithmic information content

Uncertainty

Structural, behavioral, evaluative, nested

Automated conflict avoidance for aircraft traversing airspace boundaries at different and changing altitudes and speeds, avoiding weather, considering all stakeholders have varying financial interests...

Little guidance for systems engineering
Addressing Complexity in SoSs

Source: SEBOK Wiki