Complexity in a Systems Engineering Context

Sarah A. Sheard, Ph.D.
Carnegie Mellon University Software Engineering Institute
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 \hspace{1cm} \(^{(2)}\) American Heritage Dictionary
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
- 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)

### 33 Definitions (Young, Farr, and Valerdi 2010)

<table>
<thead>
<tr>
<th>#</th>
<th>Type of Complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Hierarchical/Structural (# levels)</td>
</tr>
<tr>
<td>2</td>
<td>Configuration Complexity</td>
</tr>
<tr>
<td>3</td>
<td>Complicatedness/ Functional Complexity</td>
</tr>
<tr>
<td>4</td>
<td>Subjective Complexity</td>
</tr>
<tr>
<td>5</td>
<td>Statistical Complexity</td>
</tr>
<tr>
<td>6</td>
<td>Algorithmic/Deterministic Complexity</td>
</tr>
<tr>
<td>7</td>
<td>Aggregate Complexity (interrelationships)</td>
</tr>
<tr>
<td>8</td>
<td>Project Complexity (organizational and technological)</td>
</tr>
<tr>
<td>9</td>
<td>Project complexity (assembly, system, array)</td>
</tr>
<tr>
<td>10</td>
<td>Product Complexity (physical)</td>
</tr>
<tr>
<td>11</td>
<td>Structural Organizational Complexity</td>
</tr>
<tr>
<td>12</td>
<td>Structural IT Complexity</td>
</tr>
<tr>
<td>13</td>
<td>Dynamic Organizational Complexity</td>
</tr>
<tr>
<td>14</td>
<td>Dynamic IT Complexity</td>
</tr>
<tr>
<td>15</td>
<td>Inter-Component Complexity (can grow exponentially)</td>
</tr>
<tr>
<td>16</td>
<td>Interface Complexity (by component)</td>
</tr>
<tr>
<td>17</td>
<td>Implementation Complexity (e.g. code)</td>
</tr>
<tr>
<td>18</td>
<td>System-level Complexity (emergent)</td>
</tr>
<tr>
<td>19</td>
<td>Structural Complexity (design and structure, persistent)</td>
</tr>
<tr>
<td>20</td>
<td>Conceptual Complexity (psychological)</td>
</tr>
<tr>
<td>21</td>
<td>Computational Complexity (algorithms)</td>
</tr>
<tr>
<td>22</td>
<td>Structural/Combinatorial Complexity</td>
</tr>
<tr>
<td>23</td>
<td>Behavioral Complexity (unpredictability)</td>
</tr>
<tr>
<td>24</td>
<td>Nested Complexity (technical/socio-technical)</td>
</tr>
<tr>
<td>25</td>
<td>Evaluative Complexity (multiple stakeholder viewpoints)</td>
</tr>
<tr>
<td>26</td>
<td>Static Complexity</td>
</tr>
<tr>
<td>27</td>
<td>Dynamic Complexity</td>
</tr>
<tr>
<td>28</td>
<td>Social-Political Complexity</td>
</tr>
<tr>
<td>29</td>
<td>Technical Complexity (Systems Integration based)</td>
</tr>
<tr>
<td>30</td>
<td>Programmatic Complexity (Systems Integration based)</td>
</tr>
<tr>
<td>31</td>
<td>Configuration Complexity (Systems Integration based)</td>
</tr>
<tr>
<td>32</td>
<td>Operational Complexity (Systems Integration based)</td>
</tr>
<tr>
<td>33</td>
<td>Organizational Complexity (Systems Integration based)</td>
</tr>
</tbody>
</table>
One Chart (I will walk through this)
Complexity as an Adjective, not a Noun
Environment Elements (1)

Environment is an ongoing system: *the Way Things Are*

- Has many stakeholders
- Some of whom fund the system development effort

Has perceived problems

- Solution envisioned
- Solution and budget assigned to organization

Intend to provide desired intervention in *the Way Things Are*
Project Elements(1) and (2)
Environment Elements (2)
System Activities

Project
- System has many elements
  - Elements are different (diverse)

Technical System
- Elements have structure and behavior
- The elements are interconnected
  - In a possibly unknown way

Environment
- Many paths through the system
- System behavior emerges from connected elements
  - Of many levels
    - In a pattern
    - In a structure

Cognition
- Emergence is often surprising
Cognitive Activities

- **Project**: People are cognitively limited -> Uncertainty understanding environment, system, behavior
- **Technical System**: Uncertainty predicting changes in environment, system, behavior -> Uncertainty and unpredictability cause risk
- **Environment**: Confusion, uneasiness
- **Cognition**: Multiple people work together -> Attempt to reduce risk
- **Cognition** (end point): People analyze with modeling and cognitive support tools
Recap: A System Lifecycle?
<table>
<thead>
<tr>
<th>#</th>
<th>Type of Complexity</th>
<th>#</th>
<th>Type of Complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Hierarchical/Structural (# levels)</td>
<td>18</td>
<td>System-level Complexity (emergent)</td>
</tr>
<tr>
<td>2</td>
<td>Configuration Complexity</td>
<td>19</td>
<td>Structural Complexity (design and structure, persistent)</td>
</tr>
<tr>
<td>3</td>
<td>Complicatedness/ Functional Complexity</td>
<td>20</td>
<td>Conceptual Complexity (psychological)</td>
</tr>
<tr>
<td>4</td>
<td>Subjective Complexity</td>
<td>21</td>
<td>Computational Complexity (algorithms)</td>
</tr>
<tr>
<td>5</td>
<td>Statistical Complexity</td>
<td>22</td>
<td>Structural/Combinatorial Complexity</td>
</tr>
<tr>
<td>6</td>
<td>Algorithmic/Deterministic Complexity</td>
<td>23</td>
<td>Behavioral Complexity (unpredictability)</td>
</tr>
<tr>
<td>7</td>
<td>Aggregate Complexity (interrelationships)</td>
<td>24</td>
<td>Nested Complexity (technical/socio-technical)</td>
</tr>
<tr>
<td>8</td>
<td>Project Complexity (organizational and technological)</td>
<td>25</td>
<td>Evaluative Complexity (multiple stakeholder viewpoints)</td>
</tr>
<tr>
<td>9</td>
<td>Project complexity (assembly, system, array)</td>
<td>26</td>
<td>Static Complexity</td>
</tr>
<tr>
<td>10</td>
<td>Product Complexity (physical)</td>
<td>27</td>
<td>Dynamic Complexity</td>
</tr>
<tr>
<td>11</td>
<td>Structural Organizational Complexity</td>
<td>28</td>
<td>Social-Political Complexity</td>
</tr>
<tr>
<td>12</td>
<td>Structural IT Complexity</td>
<td>29</td>
<td>Technical Complexity (Systems Integration based)</td>
</tr>
<tr>
<td>13</td>
<td>Dynamic Organizational Complexity</td>
<td>30</td>
<td>Programmatic Complexity (Systems Integration based)</td>
</tr>
<tr>
<td>14</td>
<td>Dynamic IT Complexity</td>
<td>31</td>
<td>Configuration Complexity (Systems Integration based)</td>
</tr>
<tr>
<td>15</td>
<td>Inter-Component Complexity (can grow exponentially)</td>
<td>32</td>
<td>Operational Complexity (Systems Integration based)</td>
</tr>
<tr>
<td>16</td>
<td>Interface Complexity (by component)</td>
<td>33</td>
<td>Organizational Complexity (Systems Integration based)</td>
</tr>
<tr>
<td>17</td>
<td>Implementation Complexity (e.g. code)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
All these things change over time

All these things have information (data)
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

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

Technical characteristics

Objective complexity

Cognitive characteristics

Subjective complexity
- Uncertain
- Risky
- Difficult to understand
- Difficult to predict
- Frustrating
- Uncontrollable
- Costly
- Obsolete when built
- Unclear cause/effect

"Perceived" complexity

www.incose.org/symp2019
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