Modeling Project Complexity

John Seel
The George Washington University
Ph.D Candidate
Senior Scientist, NSWC Dahlgren

Shahram Sarkani, Ph.D., P.E
Advisor
The George Washington University

Thomas Mazzuchi, D.Sc.
Advisor
The George Washington University

Agenda

- Review of Problem
- Relevance to Today’s System Engineers
- State of Literature
- Defining System Complexity
- A Potential Approach for Describing System Complexity

The Problem

- Despite 40+ years of systems engineering there is:
  - No accepted definition of a "Complex System" [1]
  - As a result of the above, no acceptable way to model that complexity
- Some baseline literature found describing problem
  - Limited work in area, so ripe for exploitation
  - Increasing interest as general project complexity increases

Relevance

- Systems continue to grow in complexity
  - Success rates of new projects are not encouraging [2]
  - It may be impossible to fully specify a system
  - Corporations, Governments and Academia need to balance resources
- The ability to directly compare project complexity will improve:
  - Resource allocation
  - Evaluation of project risk
  - Ability to apply quantitative methods to compare projects from many sources, domains

State of Literature (1)

- Complexity of Systems Increasing [1,2,3,4]
- Many qualitative descriptions of complex systems
- Few quantitative approaches to a taxonomy or ontology to describe Complex Systems - COSYSMO is one exception
- Evolution of complex projects have driven new methods of requirements management, and newer methods may be necessary [5,6]

State of Literature (2)

- Many methods of Controlling or Mitigating Complexity via Requirements Engineering have been attempted:
  - Structured Analysis and Design Technique [7]
  - Partitioning [3,8]
  - Separation of Concerns [9]
  - Expert systems with formal syntaxes [10 ,11]
- None of these actually define system complexity
  - Some of these actually start a practitioner down the path of design before fully understanding the system

Potential Approach

- Create and define a generic model of systems complexity
- Describe factors based upon:
  - External Interfaces
  - Internal Interfaces
  - Maturity
  - Documentation Requirements
- Define numeric exponents of critical factors in complexity
  - Quantifiable
  - Similar in concept to O Notation from Computer Science
  - Provide an order of magnitude measure – Point Estimator

Possible Method

- Develop baseline approach as 4-tuple
  - $e^w \; i^x \; m^y \; d^z$
- Use case studies to examine projects of varying (known) complexities
  - Confirm that these are the right factors
  - Begin to develop proper weighting
    - Real option analysis may be of assistance
    - Utility analysis for non-quantitative data

System Complexity and Project Complexity are not the same thing

Defining System Complexity

External Interfaces:
Number of interfaces
Maturity of Interfaces
Team Experience with relevant Interfaces

Internal Interfaces:
Number of interfaces
Maturity of Interfaces
Team Experience with relevant Interfaces

Maturity:
Legacy System – no Documentation
Legacy System – limited Documentation
Legacy System – full Documentation
New Development – architecture understood
New Development – limited architecture
New Development – R&D

Documentation Requirements:
Agile
Defined, but open (SRS, SDS, etc)
Formal (Hatley Pirbhai, CORE)
Full Architecture (DoDAF,etc)
Legal
Safety

Other affecters (perceived or real)

Point Estimation of Complexity
$E^I I^J M^K D^L$

Tools for Defining Factors and Weightings

- Multidisciplinary Approach Required
- Computer Science probably leads field in this area
  - Metrics including Cyclomatic Complexity, Efferent Couplings, Lack of Cohesion of Methods, etc
- Systems Engineering approaches include those discussed as well as work from Hatley/Pirbhai, various architectural methodologies, etc

Potential Advantages of Approach

- Ability to directly compare resources across projects of differing types
  - Do projects with similar exponents have similar resources?
  - Predicting required resources from model
  - Have projects of a given complexity historically succeeded?
- Ability to graph visually the measures of complexity and compare graphs to look at indicators that may impact success/failure

Path Ahead

- Develop a generic model similar to O-Notation that can be used to quickly describe complexity of a project
  - Desired goal is a point estimator vice “the answer”
  - Useful tool for scoping project level of effort
  - Significant future refinement/growth required
- Continue to refine scope and focus of effort
- In the absence of any tools, a simple tool may be sufficient

Questions and Discussion

Backup Slides

About the Co-Authors

- **Dr. Thomas Mazzuchi** received a B.A. (1978) in Mathematics from Gettysburg College, Gettysburg, PA, a M.S. (1979) and a D.Sc. (1982), both in Operations Research from the George Washington University (GWU), Washington DC. Currently, he is a Professor of Engineering Management and Systems Engineering in the School of Engineering and Applied Science at GWU. He is also GWU's Chair of the Department of Engineering Management and Systems Engineering where he has served as the Chair of the Operations Research Department and as Interim Dean of the School of Engineering and Applied Science. He has been engaged in consulting and research in the area of reliability, risk analysis, and quality control for over twenty years.

- **Shahram Sarkani, Ph.D., P.E**., is Professor of Engineering Management and Systems Engineering at The George Washington University (GWU), and he has been a member of the faculty since 1986. His administrative appointments at GWU include chair of the Civil, Mechanical, and Environmental Engineering Department (1994-1997); Interim Associate Dean for Research and Development for the School of Engineering and Applied Science (1997-2001); and Faculty Adviser and Academic Director of EMSE Off-Campus Programs (since 2001). In his current role, he designs and administers world-wide off-campus programs in 20 locations with over 1,000 graduate students pursing Master of Science and Doctor of Philosophy degrees in several areas within the fields of systems engineering and engineering management.

Literature (1)

1. Eriksson, M., & Lillieskold, J. (2002). *Project management competence requirements when procuring complex systems.* Paper presented at the Engineering Management Conference, 2002. IEMC '02. 2002 IEEE International.

2. Helbrough, B. (1995). Computer assisted collaboration -- the fourth dimension of project management? *International Journal of Project Management, 13*(5), 329-333 Alford, M. (1994). *Attacking requirements complexity using a separation of concerns.* Paper presented at the Requirements Engineering, 1994., Proceedings of the First International Conference on.

3. Cukic, B., Ammar, H. H., & Leteef, K. (1998). *Identifying high-risk scenarios of complex systems using input domain partitioning.* Paper presented at the Software Reliability Engineering, 1998. Proceedings. The Ninth International Symposium on.

Literature (2)

4. Williams, T. M. (1999). The need for new paradigms for complex projects. *International Journal of Project Management, 17*(5), 269-273. d

5. Heninger, K. L. (1980). Software Requirements for Complex Systems. *IEEE Transactions on Software Engineering, SE-6*, 11.

6. Laufer, A., Denker, G. R., & Shenhar, A. J. (1996). Simultaneous management: The key to excellence in capital projects. *International Journal of Project Management, 14*(4), 189-199.

7. Ross, D. T., & Schoman, K. E. J. (1977). Structured Analysis for Requirements Definition. *IEEE Transactions on Software Engineering, 3*(1), 6-15.

8. Nikula, U., & Sajaniemi, J. (2005). *Tackling the complexity of requirements engineering process improvement by partitioning the improvement task.* Paper presented at the Software Engineering Conference, 2005. Proceedings. 2005 Australian.

Literature (3)

9. Alford, M. (1994). *Attacking requirements complexity using a separation of concerns.* Paper presented at the Requirements Engineering, 1994., Proceedings of the First International Conference on.

10. Edwards, M. L., Flanzer, M., Terry, M., & Landa, J. (1995). *RECAP: a requirements elicitation, capture and analysis process prototype tool for large complex systems.* Paper presented at the Engineering of Complex Computer Systems, 1995. Held jointly with 5th CSESAW, 3rd IEEE RTAW and 20th IFAC/IFIP WRTP, Proceedings., First IEEE International Conference on.

11. White, S., & Edwards, M. (1995). *A requirements taxonomy for specifying complex systems.* Paper presented at the Engineering of Complex Computer Systems, 1995. Held jointly with 5th CSESAW, 3rd IEEE RTAW and 20th IFAC/IFIP WRTP, Proceedings., First IEEE International Conference on.