Project Management and Systems Engineering:

The Relationship of Competency and Complexity

Kafi A. Joseph

12/14/2013

Project or Program Managers and Systems Engineers have distinct roles and competencies as well as overlapping roles and competencies. However, as the complexity of a system increases the roles and competencies of the systems engineer become more important and critical to project / program success. In this paper, we will review project management and systems engineering competencies to identify common ground as well as discern where and why the roles diverge. The foundation of this premise rests primarily in the NASA Competency Framework. System complexity shall be derived from the requisite experience that is required to manage larger and more comprehensive projects.

K. Joseph

I. Introduction

Project or Program Managers and Systems Engineers have distinct roles and competencies as well as overlapping roles and competencies. However, as the complexity of a system increases the roles and competencies of the systems engineer become more important and critical to project / program success. In this paper, we will review project management and systems engineering competencies to identify common ground as well as discern where and why the roles diverge. The foundation of this premise rests primarily in the NASA Competency Framework. System complexity shall be derived from the requisite experience that is required to manage larger and more comprehensive projects. Finally, the professional organizational views on program / project management and systems engineering will be reviewed.

II. Roles and Competencies of a Project Manager and Systems Engineer

The National Aeronautical and Space Administration (NASA) is responsible for planning, coordinating and executing some of the most complex programs and projects in the world. As a result, the organization has found it prudent to develop an agency-wide professional development program for project management and systems engineering. The *NASA Project Management and Systems Engineering Competency Framework* offers a detailed description of the skills, behaviors, actions, and experiences that demonstrate proficiency in each competency at four career levels ranging from project team members to managers of programs or very large projects (National Aeronautical and Space Administration, 2012, p. 1). According to the framework there are *five* distinct project management competency areas, *three* distinct systems engineering competency areas and *five* competency areas common to both roles (Figure 1). Meanwhile, Systems Engineering is a sub-competency under the Project Management Competencies, there is no sub-competency in the Systems Engineering Competencies which speaks to project management.

	Project Management		Common Competencies		Systems Engineering Competencies
PM 1.0	***Project Conceptualization***	***C 1.0***	***NASA Internal and External Environments***	***SE 1.0***	***System Design***
PM 1.1	Project Proposal	C 1.1	Agency Structure, Mission, and Internal Goals	SE 1.1	Stakeholder Expectation Definition
PM 1.2	Requirements development and Management	C 1.2	NASA Procedures and Guidelines	SE 1.2	Technical Requirements Definition
PM 1.3	Acquisition Management	C 1.3	External Relationships	SE 1.3	Logical Decomposition
PM 1.4	Project Planning	***C 2.0***	***Human Capital Management***	SE 1.4	Design Solution Definition
PM 1.5	Cost-Estimating	C 2.1	Staffing and Performance	***SE 2.0***	***Product Realization***
PM 1.6	Risk Management	C 2.2	Team Dynamics and Management	SE 2.1	Product Implementation
PM 1.7	Earned Value Management	***C 3.0***	***Security, Safety, and Mission Assurance***	SE 2.2	Product Integration
PM 2.0	***Resource Management***	C 3.1	Security	SE 2.3	Product Verification
PM 2.1	Budget and Full Cost Management	C 3.2	Workplace Safety	SE 2.4	Product Validation
PM 2.2	Capital Management	C 3.3	Safety and Mission Assurance	SE 2.5	Product Transition
PM 3.0	***Project Implementation***	***C 4.0***	***Professional and Leadership Development***	***SE 3.0***	***Technical Management***
PM 3.1	Systems Engineering	C 4.1	Mentoring and Coaching	SE 3.1	Technical Planning
PM 3.2	Contract Management	C 4.2	Communication	SE 3.2	Requirements Management
PM 4.0	***Project Closeout***	C 4.3	Leadership	SE 3.3	Interface Management
PM 4.1	Stakeholder Management	C 4.4	Ethics	SE 3.4	Technical Risk Management
PM 4.2	Technology Transfer and Commercialization	***C 5.0***	***Knowledge Management***	SE 3.5	Configuration Management
PM 5.0	***Program Control and Evaluation***	C 5.1	Knowledge Capture and Transfer	SE 3.6	Technical Data Management
PM 5.1	Tracking / Trending of Project Performance	C 5.2	Knowledge Sharing	SE 3.7	Technical Assessment
PM 5.2	Project Control			SE 3.8	Technical Decision Analysis
PM 5.3	Project Review and Evaluation				

Figure 1: NASA PM and SE Competency Framework

a. Similarities and Inherent Differences in the PM and SE Focus

While there are some broad similarities between project management (PM) and systems engineering (SE) competencies in the NASA model, it is clear that within NASA the lines of focus are programmatic for the PM and technical for the SE. Comparison of similarly labeled sub-competencies reveal the programmatic and technical divergence of the PM and SE roles; specifically the sub-competencies that address requirements definition, stakeholder interaction and risk management. The PM role in requirements definition is described as "…developing project requirements [using] functional analysis, decomposition, and allocation; …finalizing requirements into the baseline." (National Aeronautical and Space Administration, 2012, pp. 6-7) while the SE role is to "…transform base-lined stakeholder expectations into unique, quantitative, and measureable technical requirements… defining the measures of performance (MOPs) for each MOE, and defining the appropriate technical performance measures (TPMs) by which technical progress will be assessed." (National Aeronautical and Space Administration, 2012, p. 29)

Then there is the PM role in stakeholder management which "…includes identifying, soliciting, executing, and planning interrelationships with those individuals and organizations that are actively involved in the project, who exert influence over the project and its results, or whose interests may be positively or negatively affected as a result of project execution or completion." (National Aeronautical and Space Administration, 2012, p. 20) as compared to SE stakeholder expectation definition and management which "…addresses the ability to elicit and define the stakeholder's expectations through the use of cases, scenarios, and operational concepts…must identify the stakeholders and establish support strategies for them. Setting measures of effectiveness (MOEs), validating stakeholder expectation statements, and obtaining commitments from the customer and other stakeholders…" (National Aeronautical and Space Administration, 2012, p. 28)

Finally PM risk management is outlined as "…ability to identify and analyze risk and its impact; develop and implement strategies for risk mitigation; track risk; and implement continuous risk management plans. It also involves communicating risk information to all project/program levels." (National Aeronautical and Space Administration, 2012, p. 12) SE risk management is "…is achieved through regular examination of the risks of technical deviations from the plans…identifying potential technical problems before they occur [m]onitoring the status of each technical risk and implementing technical risk mitigation and contingency action plans when applicable thresholds have been triggered…" (National Aeronautical and Space Administration, 2012, p. 45)

In each of these cases programmatic activities are prevalent in the PM role to include the performance of the program/project and less details about the product or system. Conversely, the SE has very technical activities clearly tied to the performance of the product or system.

III. <u>System Complexity</u>

a. Definition of Complexity

System complexity is often blamed for many problems in system development, but there exists no single definition, framework or measure for complexity in the systems engineering community. The most basic definitions of complex consider it to be:

- a whole made up of complicated or interrelated parts (Merriam-Webster)
- consisting of interconnected or interwoven parts; composite (The Free Dictionary)
- consisting of many different and connected parts (Oxford Dictionaries)

The consistent aspects of the definitions are the interrelated, interconnected or interwoven pieces. In most of the literature comparing complex and simple systems the main difference between the two types of systems lie with these same interrelations, interconnectedness and interwoven pieces, without these interrelations, systems are generally not considered complex or "[difficult] to analyze or understand". (The Field of Complex Systems, 2003, p. 1)

A recent study of reviewed over 50 pieces of literature, articles and artifacts about complex systems The review ranged from coverage of academia to governmental applications and the authors proposed a framework to address the complexity of systems engineering. This framework reduces the types of complexity to six. Specifically, there are "… three types of structural complexity (size, connectivity, and architecture), two types of dynamic complexity (short-term and long-term), and one additional type, socio-political complexity." (Sheard & Mostashari, 2009, p. 1) The basis of understanding and interpreting the framework is the study's simplification of that which ultimately makes systems complex – the interrelated, interconnected elements and subsequent interfaces between those elements. The terms "elements" and "interfaces" are categorized as "things" and "relationships", respectively. Table 1 below (Sheard & Mostashari, 2009, p. 4) outlines other "things" and "relationships" from applicable fields like project management, process engineering and software development, which may be encountered while conducting systems engineering activities. The "things" and "relationships" depicted are considered equivalent.

Thing	Relationship	Applicable Field
Elements, Components, Systems, or Subsystems	Interfaces	Systems Engineering
Tasks	Dependencies	Project Management, PERT
Process Activities	Sequence of Activities	Process Engineering
Stocks	Flows	Systems Dynamics
Nodes	Links	Network Science
Nodes	Edges	Software Complexity
Modules	Messages	Software Development
Entities	Relationships	Systems Analysis
People	Connections	Social Networking

Table 1: Equivalence of "Things" and "Relationships"

These examples of "things" and "relationship" ensures the application of the framework is easily relatable where the term "things" can be replaced in the framework by its most suitable version. Table 1 is not all inclusive; however, domain knowledge of other equivalent "things" and "relationships" make the

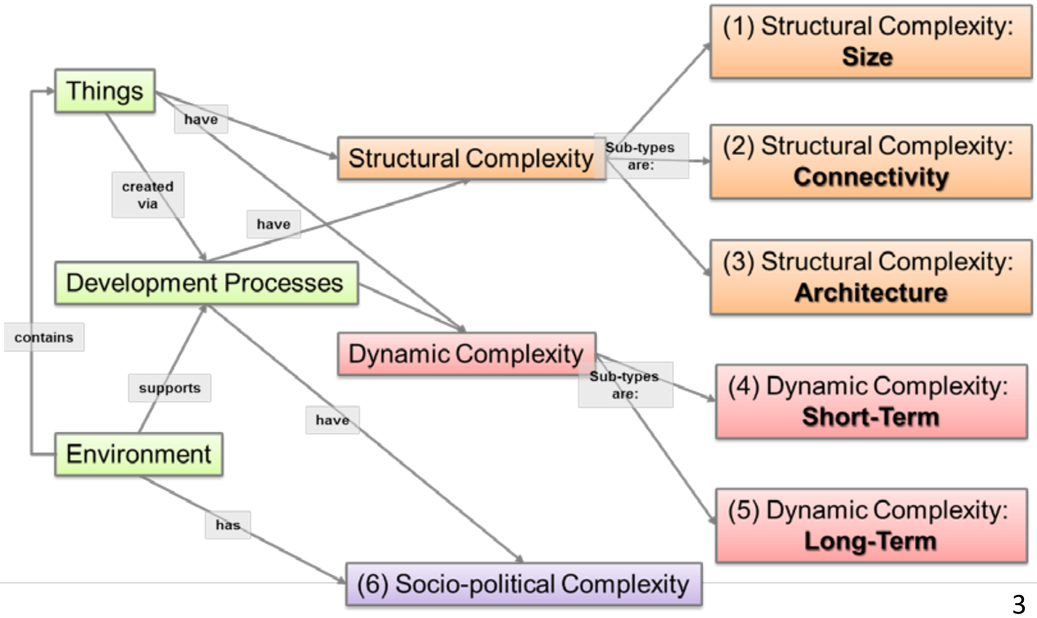

Figure 2: Complexity Framework

applicability of the framework accessible to most SE practitioners. The complexity types depicted in Figure 2 demonstrates "…how the various types of complexity interrelate. Green items (on the left) are items whose complexity is at issue, namely systems (artifacts), the processes used to develop them, and the environment. Both systems and development processes exhibit both structural and dynamic complexity. Structural complexity (orange) has three subtypes…Dynamic complexity (pink) is split into short- and long-term...Socio-political complexity (lavender) applies primarily to the environment and development processes rather than to things, although particularly in systems of systems, this plays a role in the function of the system itself." (Sheard & Mostashari, 2009, p. 5) From this description a reasonable conclusion is that "things" are inherently complex because "things" can have more than one associated complexity type 1) "Things" are created via "development process" which have more than one associated complexity; making the "thing" and "process" more complex and 2) "Things" are contained in "environments" which has an associated complexity, also making the "thing" more complex

b. Competency and Complexity

Upon determining the types of complexity aligned to the "things", the items whose complexity are at issue, the complexity decomposition can be conducted and the complexities measured at the sub-type level. Sheard and Mostashari's complexity framework includes examples and proposed measures of the six types of complexity. These examples and proposed measures make it easy to determine which level of SE can perform or lead activities based upon complexity.

Each of the competencies in NASA's Competency Framework has levels aligned and representative actions to be taken at each of the levels to ensure program success. While each competency and sub-competency contains very specific and correlating representative tasks, Table 2 on the following page summarizes the levels of responsibility, representative tasks and activities. Currently, these summarized representative tasks (or the more specific competency-based tasks) as seen in Table 3 can be compared to the possible measures/measurements to determine the SE competency level required.

At present NASA's competency framework relegates Level II SE to "simple" systems or subsystems and provides one-dimensional examples of "simple" systems. Since the competency framework is developed from lessons learned over decades of collective SE experience, NASA should explore and expand upon Sheard and Mostashari's complexity framework to further define and align the SE Competency Levels with ranges of complexity measures. For example, instead of "no more than two simple internal interfaces", NASA should conduct an historical review of projects and systems / sub-systems led by Level II SEs. Potentially, NASA may be able to leverage more Level II SEs to lead the development of systems / subsystems with more relative Structural Complexity.

The Structural Complexities of size, connectivity and architecture appear to align to the Level II SE competencies because the proposed measures are numbers of "things". The current Level II SE competency provides examples of "simple" systems in terms of numbers. NASA can begin to assess in terms of "less complex" systems vs. "simple systems" where historical data can provide a comprehensive listing of "things" and "relationships" which can, in turn, be assessed and categorized. Existing trends for the maximum number of "things" such as interfaces, tasks, nodes, etc. successfully managed by Level II SEs will inform the assessment and could potentially lead to a tiered system for the Level II Competencies, or the insertion of a level for "less complex" system management.

Dynamic and Socio-Political Complexities will likely remain at the Level III and IV SE Competencies, respectively. The proposed measures for Dynamic and Socio-Political Complexities are less straight forward. Instead of being represented as numbers of "things" these measures are time-bound in some instances and may require models or tools to determine the relative complexity.

Type	Subtype	Example	Possible Measures	Why complex?	Problem Caused?
1 Structural Complexity	Size (# elements, instances, #types elements) -of development process	People, groups, units, computer nodes -SE tasks	# items (instances) # types of unique items # development tasks	Many items means harder to track. (Some sources consider this "complicated" not complex)	Long list of elements, some won't be on time - Costly to staff all tasks
2 Structural Complexity	Connectivity (# connections, types, strengths) -of development process	Number or density of connections, data/ control/ physical, strong/ medium/ weak/ none "Static emergence" -Dependency links	# connections, density of connections (binned bystrength?) # loops or threads per McCabe software complexity.	Everything connects to everything else. Hard to decompose. Determines network connectivity, path length, feedback loops	Emergent properties (intended/ desirable/ undesired) Cascading failures Difficult to observe, describe, control, predict breakage -Program may not converge
3 Structural Complexity	Architecture (Patterns, chunkiness of connections, boundaries)	Trees, layers, networks, teams Chunks, fragmentation, "non-holonomic constraints"	Measures similar to texture measures, such as size distribution function (Moskowitz and Jacobs 1975); (Kaye, Junkala et al. 1998) or Boundary fractal dimension (Wettimuny and Penumadu 2003)	Chunks of order have different properties from other chunks or from the stew of complexity	Averaging across different types gives incorrect conclusions
4 Dynamic Complexity	Short term (Nonlinearity, dynamic emergence, sudden rapid change in system behavior— butterfly effect) -development system behavior	Homeostasis, Time patterns, Feedback. -Execution of development system (building of the product) is inherently dynamic	In a system dynamics or agent-based model, predict characteristics of changes such as frequency and size (or consequence), and derive "change exposure" measures? - # Paths, deviations from waterfall, dependencies? PERT C measures	Difficult to predict, control, understand, or communicate. Feedback and time delays. -tasks may happen out of sequence	Things can change suddenly and unpredictably. Hard to estimate. Typical linear and gaussian- based assumptions fail.
5 Dynamic Complexity	Long term (changes in # and types of things and relationships)	Origin of complex sys- tems, growth, CAS cycle, self-organization, adaptation, learning, fitness landscape changes.	ESE Profiler tool results Possibly: measures of resilience	Evolution is long-term emergence. What is learned changes short-term and long-term behavior.	Adaptation is hard to predict. Difficult to engineer system qualities ("ilities") at end of life because both system & context change
6 Socio-political Complexity	Social and Political (Human cognitive limitations, multiple stakeholders, global context, environmental sustainability, economics) -"Coop-etition," supplier chain depth, distributed development	Objectives multiple, soft, value-laden Stakeholders multiple, with multiple perspec- tives, multiple approvals required. Pluralism (Social chunk- iness; non-holonomic) Sociological aspects of teams and organizations, Diverse operational environments, diffuse boundaries	ESE Profiler, sociological measures of group and organizational coherence, team measures, political measures such as size of constituency. Scale of effort in terms of number of users and user types.	Individuals ambivalent and change their desires over time. Team dynamics and organizational dynamics can scuttle technical concerns. Politics can determine whether a system is built at all. Policies are resisted.	Goals change, acceptability of developed system changes, system requirements change, funding is cut, system may not be developable in time required.

Table 2: Complexity Types, Sub-Types, Examples and Possible Measures

Competency Level	Representative Tasks	Activities
I **Techncial Engineer**	Performs fundamental and routine SE activities while supporting a Level II-IV SE as a member of a product team	• Technical Integration Knowledge • SE and Basic Project Management
II **Sub-System Lead**	Performs SE activities for a subsystem or simple project (e.g., no more than two simple internal / external interfaces, simplere contracting processes, smaller team / budget, shorter duration)	• Leadership Application • Participation in SE
III **Project Systems Engineer**	Performs as a SE for a complex project (e.g., distinct subsystems or other defined services, capabilities and associated interfaces)	• Directing • Structuring • Integration Activities
IV **Program / Center / Agency Systems Engineer**	Oversees SE activities for a program with several systems and/or establishes SE policies at the Agency or Center level	• SE Strategy Large Complex Initiatives • Management of Agency Initiatives

Table 3: NASA SE Competencies, Tasks and Activities

The structure of NASA's Competency Framework lends credence to the importance of competency as the complexity of a system increases. The requirements for ascending in competency (Levels I- IV) within the organization increases with the relative complexity of the systems being developed / led. NASA's SE Competencies are structured by: 1) competency area, 2) competencies, 3) competency elements, 4) proficiency level descriptions, 5) courses and learning activities. These terms are defined below:

- Competency Areas: what is expected of Systems Engineer personnel in terms of particular components or functions of the job.
- Competencies: overall knowledge, skills, behaviors that SEs are expected to possess / perform
- Competency Elements: describe the specific knowledge, skills, behaviors, which can be measured against established standards, can be improved via training and development activities, and correlate to performance on the job.
- Proficiency Level Descriptions: specify the knowledge/performance to be achieved in order to demonstrate successful mastery of the competency and are expressed in terms of levels.
- Courses and Learning Activities: outline the required/suggested courses and activities to obtain proficiency in the competencies by level. (NASA ACADEMY OF PROGRAM/PROJECT & ENGINEERING LEADERSHIP, p. 4)

IV. Community Views on PM and SE Roles

Earlier, this document highlighted the "Systems Engineering" sub-competency within the "Project Implementation" Competency, which is a specifically a PM competency group. The presence of this competency indicates that as a system becomes more complex, a PM would require more technical systems engineering knowledge, in addition to project/program management skills. Further literature from both the PM and SE communities substantiate this premise. As systems, and by extension projects / programs, become increasingly complex the roles of the PM and SE diverge to converge again with emphasis placed on the SE activities being critical for overall program / project success.

Despite the emphasis on technical knowledge – across the two communities there is continued emphasis on the PM and programmatics, but how technical artifacts quickly impact the programmatic success (e.g., technical risk as it relates to cost and schedule; and technical requirements as they relate to performance and scope / scope creep.

a. Programmatic-Focus for PM.

Programmatics belong with the PM and experienced SEs should serve as PM technical advisor to ensure that interdependencies of technical issues with programmatics are characterized. In cases of technically complex programs, it is advantageous to appoint a PM with a strong technical background (Van Gemert, 2013). The NASA Competency Framework is a prime example of this principle in implementation. PMs are required to have SE competencies, as most NASA projects / programs can be classified as technically complex programs.

A high-level comparison of the PMBok, DAU Handbook and INCOSE Handbook outlines reiterates the programmatic-focus of the PM community. There are 40+ PMBoK Processes, with considerably less in the others: 17 Processes in the DAU Handbook Chapter 4 under Technical Management and Technical Processes and seven project planning processes in the INCOSE Handbook Chapter 5, along with the Technical and Enterprise Processes of Chapters 4 and 6 of INCOSE.

b. Systems Engineering Competencies & Tasks Yield Greatest Impact on Program Success

Considering the comparison of the DAU and INCOSE Handbook Processes to the PMBoK, emphasis on the technical aspects of project planning are evident and have become more prevalent in the PM community. Where a cultural barrier once grew between practitioners of systems engineering and of program management and managers developed the mindset that their work activities were separate from each other, rather than part of an organic whole (Langley, Robitaille, & Thomas, 2011), there has come a paradigm shift.

The Planning Phase of a project is now considered the most critical phase for SE involvement on the project leadership team. It is where "SEs identify technical risks, manage and derive requirements, align the technical baseline with the project baseline, and translate technical issues into actionable business cases that the project manager can use to make critical business decisions." (Van Gemert, 2013)

V. Conclusion

PM and SE Roles and responsibilities will remain distinct with overlapping competency areas. There are considerable benefits to overlapping competencies as it relates to organizational or domain knowledge. However, the magnitude and breadth required of programmatic processes, as exemplified by the PMBoK processes juxtaposed with the depth required for technical processes, particularly in the development of increasingly complex systems, ensures that the roles will continue to remain distinct. The areas of overlap will vary with the complexity and types of project and products; and SE will continue to hold a position of importance, not only with system complexity, but also in ensuring programmatic success.

Works Cited

Langley, M., Robitaille, S., & Thomas, J. (2011, September). Toward a New Mindset: Bridging the Gap Between Program Management and System Engineering. *PMI Netowork*. Retrieved August 15, 2013, from http://www.pmi.org/Business-Solutions/~/media/PDF/Business-Solutions/Lean-Enablers/PMN0911-INCOSE.ashx

Merriam-Webster. (n.d.). Dictionary. Retrieved December 2013, from http://www.merriam-webster.com/dictionary/complex

NASA ACADEMY OF PROGRAM/PROJECT & ENGINEERING LEADERSHIP. (n.d.). *NASA.* Retrieved November 22, 2013, from NASA's Systems Engineering Competencies: http://www.nasa.gov/pdf/303747main_Systems_Engineering_Competencies.pdf

National Aeronautical and Space Administration. (2012). Project Management and Systems Engineering Competency Framework (Revision 3.0). Retrieved October 19, 2013, from http://www.nasa.gov/pdf/699790main_PM_SE-Competency_Model_rev_2012_09_24_12.pdf

Oxford Dictionaries. (n.d.). Dictionary. Retrieved December 2013, from http://www.oxforddictionaries.com/us/definition/american_english/complex

Sheard, S. A., & Mostashari, D. A. (2009). *A Complexity Typology for Systems Engineering.* Stevens Institute of Technology, Hoboken. Retrieved November 23, 2013

Stevens, R. (2006). *Engineering Enterprise Systems: Challenges and Prospects.* MITRE Corporation, McLean, VA. Retrieved November 23, 2013

(2003). The Field of Complex Systems. In Y. Bar-Yam, *The Dynamics of Complex Systems — Examples, Questions, Methods and Concepts* (pp. 1-15). Retrieved November 2, 2013, from http://www.necsi.edu/guide/DCSchapter0.pdf

The Free Dictionary. (n.d.). Dictionary. Retrieved December 2013, from http://www.thefreedictionary.com/complex

Van Gemert, D. (2013). Systems Engineering the Project. *PMI Virtual Library*. Retrieved October 19, 2013, from http://www.pmi.org/Knowledge-Center/~/media/PDF/Publications/Van%20Gemert%20White%20Paper%20for%20RM%20page.ashx