SYSTEMS
ENGINEERING
RESEARCH CENTER

The Development of *Atlas*: The Theory of Effective Systems Engineers

Technical Report SERC-2016-TR-119

December 16, 2016

Research Team:

Dr. Nicole Hutchison, Dr. Devanandham Henry, Dr. Dinesh Verma
Megan Clifford, Ralph Giffin

Stevens Institute of Technology

Dr. Art Pyster

George Mason University

Work conducted under Research Topic 154

Contract No. HQ0034-13-D-0004 Task Order 0054, RT 154

The Systems Engineering Research Center (SERC) is a federally funded University Affiliated Research Center managed by Stevens Institute of Technology.

This material is based upon work supported, in whole or in part, by the U.S. Department of Defense through the Assistant Secretary of Defense for Research and Engineering (ASD(R&E)) under Contracts H98230-08-D-0171 and HQ0034-13-D-0004.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

The Helix team would like to thank all the organizations and individuals that willingly participated in the project, offering their resources, time, and effort. This was critical to our research. Their active participation in the Helix interviews provided us data that was rich in both quality and quantity, which makes this research more valuable and useful to the participating organizations and the systems engineering community at large.

We are most grateful to the Office of the Deputy Assistant Secretary of Defense for Systems Engineering (DASD(SE)), especially Kristen Baldwin and Scott Lucero, for their continued support, without which this research would not be possible. The International Council on Systems Engineering (INCOSE) and the National Defense Industrial Association Systems Engineering Division (NDIA-SED) are both valued partners in this research and we thank them, especially Courtney Wright, David Long, Bill Miller, and Don Gelosh.

We also thank all former members of the Helix research team whose contributions have shaped our research over the years. In particular, we would like to thank Dr. Art Pyster, whose leadership and dedication since the beginning of the project were critical and instrumental to its success.

Nicole AC Hutchison

Nicole AC Hutchison

Helix Principal Investigator

EXECUTIVE SUMMARY

Atlas 1.0 is the culmination of over four years of research into what makes systems engineers effective. The key elements that play a role in effectiveness are identified in Figure 1 below. The specifics defined for each of these variables are the result of in-depth research on systems engineers. Additionally, related disciplines such as classic engineering (electrical, mechanical, software, etc.) or systems-related professions are also expected to find these materials applicable with slight tailoring.

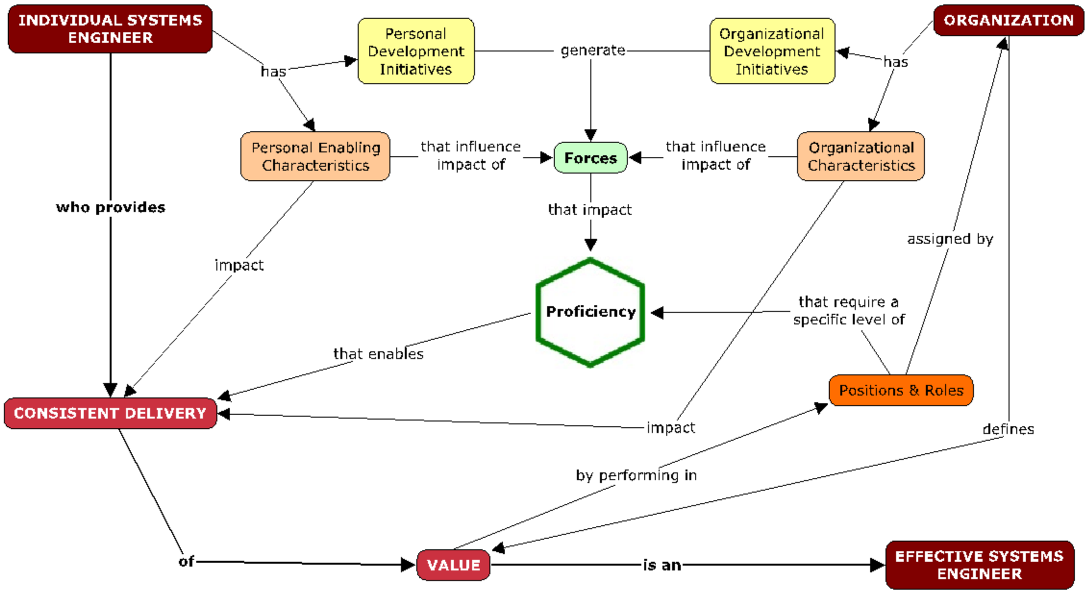


Figure 1. *Atlas 1.0* Overview

The main theme of *Atlas* is an *Individual Systems Engineer who provides Consistent Delivery of Value is an Effective Systems Engineer*. This definition hinges on Value, which is defined by the Organization in which a systems engineer is working. Value is created by working in defined positions and roles. The organization must establish the position of the systems engineer in terms of roles and responsibilities and this should align with specific levels of Proficiency – knowledge, skills, abilities – that enable a systems engineer to perform in a given position.

Both individuals and organizations may have development Initiatives; together, they generate forces – experiences, mentoring, or education and training – that impact proficiency. At the same time, personal and organizational characteristics influence the impact of forces on proficiency – positively or negatively. Both personal and organizational characteristics impact consistent delivery of value. Amidst these variables and their interactions, the challenge for the individual systems engineer and the organization is to improve the proficiency that enables consistent delivery of value to the organization.

Atlas is expected to be used in several ways: first, by individuals who wish to better understand their own proficiencies and effectiveness in the context of their organization; second, by organizations that wish to understand the current state of the effectiveness of their systems engineers; and third, by either individuals or organizations for future career planning. These use cases and recommended approaches are described within this document.

A companion document, *Atlas: The Theory of Effective Systems Engineers, version 1.0*, provides a streamlined version of *Atlas* and this document is recommended for most users.

1 BACKGROUND AND INTRODUCTION

The Systems Engineering Research Center (SERC), a University Affiliated Research Center (UARC), set up by the U.S. Department of Defense (DoD), responded to the systems engineering workforce challenges by initiating the Helix Project to investigate the "DNA" of systems engineers, beginning with those who work in defense and then more broadly. The US Deputy Assistant Secretary of Defense for Systems Engineering (DASD(SE)), the International Council on Systems Engineering (INCOSE) and the Systems Engineering Division of the National Defense Industrial Association (NDIA-SED) jointly sponsor Helix. To ensure Helix delivers the greatest value and to help Helix obtain access to the necessary data, Helix formed the Helix Advisory Panel (HAP) with representatives primarily from those three sponsor organizations. Helix has held three annual workshops with a broad set of representatives from across government, academia, and industry.

Helix is a multi-year longitudinal research project, which has gathered data from many organizations with DoD and the Defense Industrial Base (DIB) through a combination of techniques, including interviews with hundreds of systems engineers. In 2014, Helix began to reach beyond DoD and the DIB, to gather data from other types of organizations as well, including non-defense organizations in the US and non-US organizations. Version 0.25 of *Atlas* was also published in 2014. *Atlas* identifies the key variables that impact a systems engineer's effectiveness – positively or negatively – and provides, as much as possible, details on how these variables impact effectiveness.

During 2015, Helix expanded its data collection by conducting interviews with non-DoD organizations as well; matured *Atlas* into the next versions, *Atlas 0.6*; defined and analyzed the career paths of systems engineers; and did implementation trials of *Atlas*.

During 2016, the team generated *Atlas 0.6* and *Atlas 1.0. Atlas 1.0* reflects the results of analysis of in-depth interviews with 287 individuals. Most of these individuals were systems engineers, though approximately 10% of the sample was comprised of individuals who work with systems engineers – organizational leaders, classic engineers (electrical, mechanical, software, etc.), and program managers. In 2016 the Helix team also worked on implementation of *Atlas* with a number of organizations and lessons learned from those activities are captured here.

1.1 THE HELIX PROJECT

The US Department of Defense (DoD) and the Defense Industrial Base (DIB) – contractors that develop and deliver systems to the DoD – have been facing major systems engineering challenges in recent years (e.g. GAO 2008, 2011, 2012, 2013). Mission requirements are evolving and they demand ever more sophisticated and complex systems (e.g. Boehm et al. 2010; INCOSE Technical Operations 2007; Davidz 2006; Davidz and Nightingale 2007; Frank et al. 2007; INCOSE 2014); the tools, processes, and technologies that systems engineers must master keep changing more rapidly (e.g. Frank 2006); and budgets and schedules are being compressed dramatically. An additional concern is that thousands of systems engineers in the defense workforce are nearing retirement; they will take with them hundreds of thousands of staff-years of experience (DoD 2013).

Organizations have responded to these challenges in a variety of ways, such as offering extended training and education to their current workforce or systematically seeking to select specialty engineers with the potential to become systems engineers and incorporating them into the ranks of systems engineers. Unknown is whether these actions are producing the desired results because there is no common understanding of the diverse roles that systems engineers play, how they are selected and evaluated, what competencies are most important for different roles, how to evaluate effectiveness, or how experiences impact effectiveness. These and many other insights will be critical to maintaining and growing the systems engineering workforce in the US DoD and DIB.

1.2 HELIX RESEARCH QUESTIONS

There are three questions that the Helix project is intended to address:

1. What are the characteristics of systems engineers?
2. How effective are systems engineers and why?
3. What are employers doing to improve the effectiveness of their systems engineers?

It was in the pursuit of answering these questions that *Atlas* was developed. The team believes that *Atlas* sufficiently answers Question 1 by providing detailed insight into the critical skills, development paths, and values provided by systems engineers. Question is answered by *Atlas* itself, which outlines how various elements come together to generate a systems engineers' effectiveness. Question 3 has be somewhat addressed, but will be carried forward in future work.

1.3 HOW IS ATLAS DIFFERENT FROM HELIX?

Helix is the name of the overarching SERC project. Helix has been examining what makes systems engineers effective for over four years. As a project, Helix has created many different deliverables or products. The primary product of Helix is *Atlas: The Theory of Effective Systems Engineers*. This document represents *Atlas 1.0* – expected to be mature enough for individuals or organizations to use without direct help from the Helix team. It is a standalone document to detail the contents of *Atlas*.

This document does *not* contain all of the research that led to the development of *Atlas 1.0*. Instead, the detailed research results and how they led to *Atlas 1.0* are contained in the companion Helix Technical Report (SERC-2016-TR-118). Individuals or organizations that want not just to use *Atlas* but to also understand the rationale and methodology behind its development should reference the Technical Report. Several earlier published Helix papers and technical reports are also referred to throughout this report. The reader is not expected to read the earlier technical reports or any of the other Helix papers or reports, in order to understand *Atlas 1.0*.

In addition, there are tools that an individual or organization can use to support self-assessment using *Atlas*. The paper-based tools are contained in the Appendices of this report. The team has also developed more easily tailored Excel-based tools, which can be found on the Helix page of the SERC website (http://www.sercuarc.org/projects/helix/).

The relationship between Helix, *Atlas*, the Technical Reports, and the tools is illustrated in Figure 2.

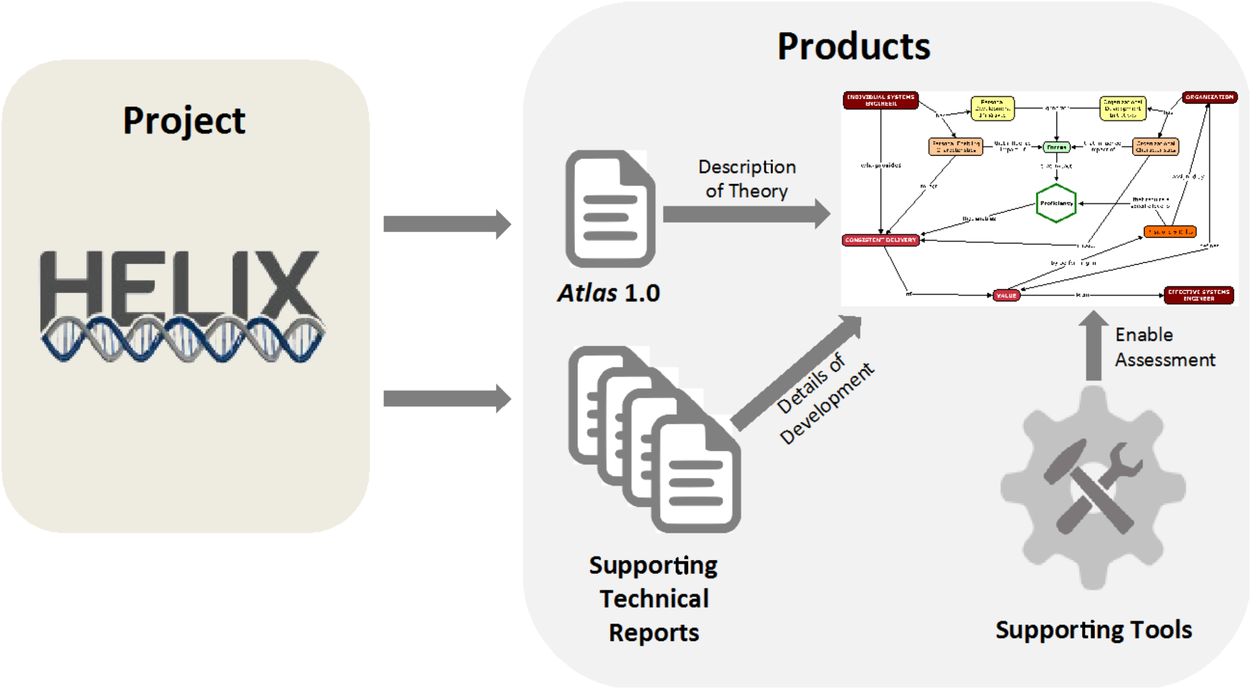


Figure 2. Relationship between Helix and *Atlas*

All of the details of *Atlas* 1.0 are contained in this report, but unlike in the *Atlas* 1.0 document, there is also an explanation of the available data for each element of *Atlas*, the analyses conducted, relevant findings, and how this shaped the development of the theory.

1.4 INCREMENTAL DEVELOPMENT OF ATLAS

The Helix project used an incremental approach to develop *Atlas*. This approach was designed to enable publication and use of aspects of *Atlas* as they became appropriately mature, while maintaining the expectation that *Atlas* would become more mature over time. The increments were:

- ***Atlas 0.25***: The first draft of *Atlas* based on work done in 2014 was published as *Atlas 0.25* in November 2014. It included key elements that explain the effectiveness of systems engineers, and a preliminary explanation of the relationships between those elements. The structure and variables of the proficiency model were also included, along with some initial analysis of career paths.
- ***Atlas 0.5***: Based on subsequent work done in 2015, *Atlas 0.5* was published in December 2015. It reflected further understanding of the elements of *Atlas* and their inter-relationships. Significant new work was done in the area of career paths and *0.5* incorporated initial efforts to use *Atlas* to assess the level of proficiency of systems engineers. *Atlas 0.5* was mature enough for an individual or an organization to use and gain valuable insights with some guidance from the Helix team.
- ***Atlas 0.6***: Was an incremental improvement to *Atlas 0.5*. It contained additional detail and analysis for areas that were less mature in 0.5, namely: mentoring, personal initiatives, and organizational initiatives. *Atlas 0.6* was not created as a stand-alone document, but rather as a supplement to 0.5.

- ***Atlas 1.0***: *Atlas 1.0* – this document – includes a more complete description of the elements of *Atlas* and their inter-relationships. *Atlas 1.0* is believed to be mature enough for independent deployment and assessment by individuals and organizations with little or no guidance from the Helix team. In addition, the frameworks presented in *Atlas 1.0* have been validated using data from outside the US DoD, and therefore is believed to be applicable to systems engineers in a variety of domains. This is intentional. Though the initial impetus for the work was based on the needs of the US DoD, the Helix team believes that a more generic framework which benefits all systems engineers, regardless of domain, is both more beneficial to the community at large and, ultimately will benefit the US DoD by setting consistent expectations for practitioners across domains.

Atlas 0.25 and *Atlas* 0.5 were mature enough for trial. The Helix team is aware of five organizations which have used some aspects of *Atlas*, primarily to assess the proficiency levels and understand the career paths of individual systems engineers within the organization. Feedback and observations from these early use exercises influenced the development of *Atlas 1.0* as published here. A glimpse into potential benefits of *Atlas* deployment, based on trials conducted in 2015 and 2016, are included in Section 6 of this report.

1.5 ABOUT THIS DOCUMENT

This technical report is written as a standalone document, presenting version 1.0 of *Atlas: The Theory of Effective Systems Engineers*. Several earlier published Helix papers and technical reports are referred to throughout this report. However, the reader is not required to read the earlier technical reports or any of the other Helix papers or reports, in order to understand *Atlas 1.0.*

Readers should note the following about the report:

- Throughout the report, the term 'Helix' is used to denote either the project or the team that performed the work in developing *Atlas*.
- The *Guide to the Systems Engineering Body of Knowledge* (SEBoK) is used across the report as the primary source of consistent terminology and definitions relevant to systems engineering. (BKCASE Editorial Board 2016)
- All insights and observations are presented only in an anonymous, aggregated manner. Individuals or organizations that participated in the Helix project are neither named nor are they identifiable from this report.

The report is organized as follows:

- Section 1 provides the necessary background and introductions to the Helix project and to *Atlas: The Theory of Effective Systems Engineers*:
- Section 2 provides the methodology for Helix. This includes the data collection and analysis approaches as well as the philosophical approach which has guided the project since its inception. Finally, section 2 provides a detailed overview of the data from Helix – the sources, demographics from the sample population, and characterization of the data.
- Section 3 provides a brief overview and introduction to *Atlas 1.0*.
- Section 4 provides all of the details of *Atlas 1.0*, including supporting data and key decisions.

- Section 5 provides the *Atlas* career path approach and the details around the career paths of systems engineers as examined to date.
- Section 6 provides insights into the use of *Atlas*, including expected uses, based on what the Helix team has learned working with organizations that have used *Atlas* in 2015 and 2016. This includes key recommendations that are highlighted elsewhere in the document.
- Section 7 provides details on planned future directions for Helix.

The Helix team believes that this technical report provides sufficient detail for any user to understand not only the details of *Atlas 1.0* but also the rationale for why the theory looks as it does.

2 HELIX RESEARCH APPROACH

This section provides detail on how the Helix project has been structured since its inception. This includes overarching methodology, details of analytic approaches, and detail about the dataset. The organization of Section 2 is as follows:

- The overarching philosophical approaches to the research (Sections 2.1-2.3),
- The detailed analysis approaches for the research (Sections 2.4 and 2.5),
- An overview of the dataset (Section 2.6), and
- Guidance on how to interpret the data, including limitations of the dataset (Section 2.7).

2.1 RESEARCH PHILOSOPHY

Helix is primarily a qualitative study, with the primary means of data collection being interviews with systems engineers. From 2012-2013, the Helix team focused on a mixed-methods approach (Creswell and Plano 2011), combining the development of basic research questions with a grounded theory approach. Grounded theory was developed in the social sciences as a method for developing theory that is grounded in data that is systematically gathered and analyzed. (Goulding 2002) This approach allows the data itself to drive points of further inquiry, guide categorization, etc.; rather than starting analysis with an existing framework, all of the data is reviewed holistically and any potential areas of interest are coded. Over time patterns emerge and these guide further data collection and analysis. The development of driving research questions makes the Helix project mixed method as opposed to pure grounded theory.

When performing initial data coding, the Helix team coded all data, not making suppositions about which data would prove “important”. The team also compared data collected against the existing literature where possible. For example, as systems engineers defined the activities that they perform, the team collected and organized the raw data but also compared it to the “Twelve Roles of Systems Engineers” defined in (Sheard 1996).

This approach still reflects the philosophy of Helix: *Atlas* 1.0 is largely a reflection of the data, using the grounded theory principles to “let the data speak”.

2.2 OVERARCHING RESEARCH METHODOLOGY

The research methodology adopted for Helix research may be considered to be a modified grounded theory based approach, employing qualitative and quantitative research methods.

During 2013 and 2014, Helix primarily focused on data collection from DoD and DIB organizations through semi-structured in-person interviews with individuals or small groups, continually refining the interview questions and process. Follow-up interviews were conducted by telephone with most of the participants. Analysis of the data to address the Helix research questions offered insights into the effectiveness of systems engineers and led to the development of an early version of *Atlas* that was published in November 2014. During 2015, data collection was expanded to organizations outside of DoD and DIB, and *Atlas 0.25* was validated and improved upon, leading to the next version, *Atlas 0.5*, published in December 2015.

The Helix project adopted a grounded theory approach because it did not presuppose any specific theory or propose any hypotheses at the start of the project. Grounded theory was developed in the social sciences as a method for developing theory that is grounded in data that is systematically gathered and analyzed (Goulding 2002). Rather than beginning with a hypothesis, the first step was data collection. This approach is unusual in engineering research, where a researcher traditionally begins with a theoretical framework that he or she applies to the phenomenon to be studied. In the Helix project, the data collected from the many semi-structured interviews were marked up with codes that were grouped into concepts, that led to the identification of constructs and categories that formed the building blocks of *Atlas*. This approach minimized any bias that might be introduced by the researchers, instead allowing the large data set collected through the Helix project to drive theory development. Having established a preliminary theory of effective system engineers and proficiency model of systems engineers, data collection and interviews conducted during 2015 focused on validating *Atlas*, and refining the theory towards developing *Atlas 1.0* in 2016.

Qualitative research aims to create or discover what things are made of, and what is created or discovered are called constructs. Qualitative research is useful for obtaining insight into situations and problems on which one has little knowledge a priori. This method is commonly used for providing in-depth descriptions of procedures, beliefs and knowledge, including the opinions of respondents about particular issues; detailed data is gathered through open-ended questions. Data collection for the Helix project and subsequent analysis of the data was primarily done employing qualitative research methods; appropriate software tools were used to support coding and identification of constructs.

Quantitative research begins once initial constructs are in hand. It attempts to gather data by objective methods to provide information about relations, comparisons, and predictions. In the context of the Helix project, quantitative research was performed once initial constructs for demographics of systems engineers, their organizations, and their career paths were established. Data was collected from their resumes, as well as through pointed questions during interviews. Quantitative analysis continues to be performed on various elements of *Atlas* that were developed based on qualitative research, particularly on the proficiency model.

2.3 HELIX RESEARCH PROCESS

The Helix research methodology discussed in the preceding section was deployed using the research process illustrated in Figure 3 below. The Helix research process consists of seven major steps:

A. Preparation for Data Collection
B. Data Collection
C. Data Analysis
D. Methodology Review
E. Theory Development
F. Publishing
G. Validation, Feedback & Deployment

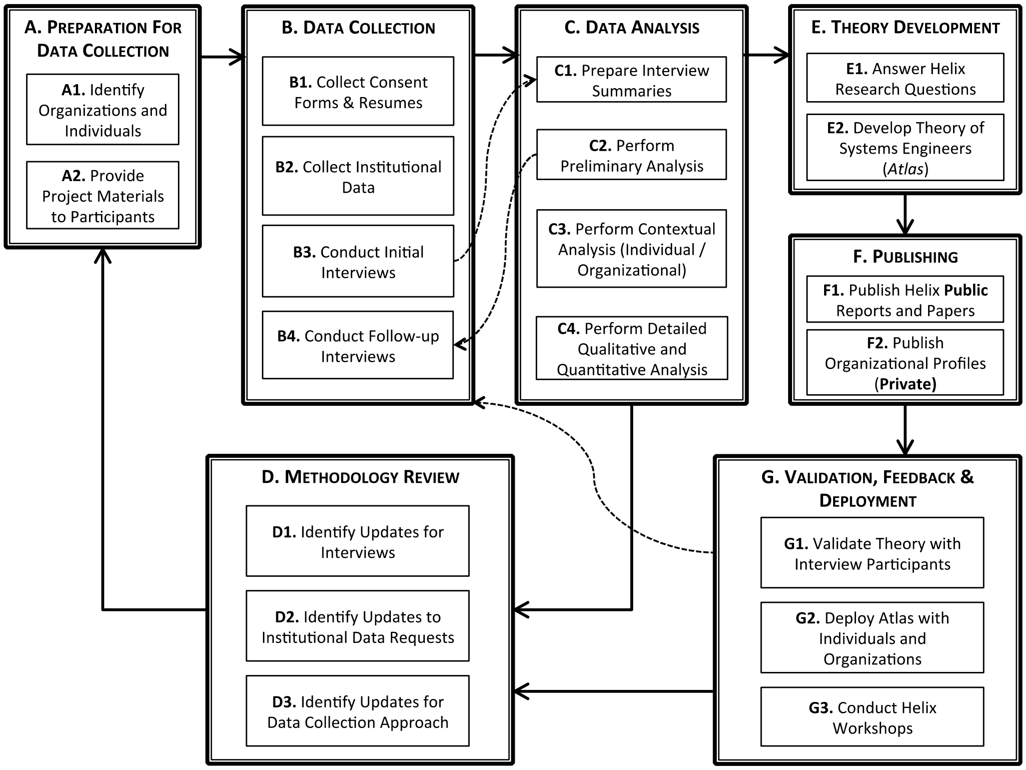


Figure 3. Helix Research Process

The focus of Helix in 2013 was on executing the loop **A-B-C-D-A** multiple times with different organizations. The loop **B-C-B** was executed a few times when follow-up interviews were conducted with some organizations. During 2014, in addition to performing the **A-B-C-D-A** loop with new organizations, steps **E-F-G** were executed that led to initial the development of *Atlas 0.25*. In 2015, much effort was concentrated in executing step **G**, as well as executing the loop **A-B-C-D-A** with commercial non-DoD organizations as well as with many participants who did not consider themselves to be systems engineers. In 2016, the primary efforts were focused on executing step **G**, which included supporting organizations as they determined how to implement *Atlas* and conducting extensive outreach with the systems engineering community. This has led to further refinement of Atlas in step **E**, leading to step **F** - the publishing of *Atlas 1.0* and this report.

2.3.1 PREPARATION FOR DATA COLLECTION (A)

Since Helix research is based on a grounded theory approach, preparation for data collection was the first step executed in the project. Initially, organizations from within the US DoD and other organizations from the DIB were identified for data collection; also, the primary focus was on systems engineers in these organizations (**A1**). As Helix progressed, in 2015 other commercial organizations from non-DoD sectors such as healthcare and information technology were identified for data collection. The latest reports and papers published from the Helix project were provided to potential interviewees (**A2**). Based on their willingness to participate in Helix interviews, the organization makes final decisions on who participates in Helix interviews. In 2016, the primary focus has been on additional publication and on assisting with the implementation of *Atlas* at several organizations. These efforts have helped the team understand where the theory needed adjustments to make it easier to utilize.

2.3.2 DATA COLLECTION (B)

The first round of data collection with an organization is typically through a site visit to the organization, where in-person interviews are conducted. Typically, there will be 2 or 3 Helix interviewers and anywhere from 1 to 6 interviewees in a single 90-minute interview session (**B3**). Following approved research protocols, a signed consent form is collected from the participants before conducting interviews; resumes are also requested from all participants (**B1**). Any available organizational data that will provide insights into the systems engineers in the organization and how they are structured within the organization are gathered before and during the site visit (**B2**). In 2015, as the project expanded to include non-DoD participants, initial interviews were conducted over telephone when the number of participants from an organization was very low. All follow-up interviews were conducted over telephone (**B4**). In 2016, there have been no additional interviews. Instead, the team has focused on analysis of the data collected and on assisting organizations with the implementation of *Atlas*. Data gathered in 2016 has been focused, therefore, on an issues identified with implementation and determination of whether those issues reflected a weakness in *Atlas* to be addressed or whether they were a reflection of the unique environment of a given organization.

2.3.3 DATA ANALYSIS (C)

The first step in data analysis is to prepare summaries of all interview sessions (**C1**). Where interviewees permit audio recording, transcripts are first created then cleaned and prepared for further analysis. If recording is not permitted, summaries are created from the notes taken during the interviews. Preliminary analysis, typically not employing significant effort on using analysis tools, is performed to quickly identify additional questions to be asked or additional data to be collected during follow-up interviews (**C2**). Since 2014, significant research effort has been put into performing contextual analysis on an individual, particularly on her career path (**C3**). Detailed qualitative and quantitative analyses, using software tools as necessary, have been performed on the large amounts of data that have been collected through Helix interviews (**C4**). These analyses make significant contributions to theory development efforts (**F**).

2.3.4 METHODOLOGY REVIEW (D)

Data collection and analysis is being performed iteratively, as Helix continues to identify and visit organizations. After any site visit and before the next one, a review is conducted to identify any updates to the interview questions or process (**D1**). While much organizational data is desired for Helix analysis, not all information is being made available within an organization in a form that may be readily shared with Helix researchers. Based on experiences with organizations, the nature and content of organizational data requested has been regularly updated (**D2**). Based on significant data analysis and theory development that was performed in 2014, the data collection approach was revised from being a semi-structured interview to being a discussion on assessing the proficiencies of individuals and analyzing their career paths (**D3**). Feedback received from individuals and organizations on the Helix reports (**G**) also influenced the updates performed in step **D**. In 2015, work on broad implementation has identified areas has been the focus. However, if additional in-depth interviews are conducted in future, the approach of reviewing findings and conducting interviews around existing frameworks is expected to remain consistent.

2.3.5 THEORY DEVELOPMENT (E)

Analysis performed on data collection during 2013 focused primarily on answering the Helix research questions at a broad level (**E1**). Since 2014, the focus of analysis has been to develop *Atlas* (**E2**). Version 0.25 of *Atlas* was published in November 2014. *Atlas 0.5* was published in December 2015 and an incremental improvement, *Atlas 0.6* was published in April 2016. Refining this theory, and packaging it for independent assessment and deployment by individuals and organizations has been the focus of research efforts in 2016. These efforts culminated in the development of *Atlas 1.0*, as reflected in this report.

2.3.6 PUBLISHING (F)

Publishing reports and papers for public consumption is a key objective for Helix research (**F1**). All results and observations reported in Helix publishing are done in an anonymous aggregated manner. Nothing published by Helix is traceable to any particular individual or to an organization. Organizations may choose to reveal their participation in the Helix project, but they are not listed in any Helix report. In addition, peer-reviewed conference and journal papers continue to be published for wide dissemination of Helix results. While some form of an organizational profile is created as part of internal Helix analysis, in some rare cases, a private report is provided to participating organizations upon request to support their systems engineering workforce development efforts (**F2**).

A complete list of Helix-related publications can be found in Appendix A.

2.3.7 VALIDATION, FEEDBACK & DEPLOYMENT (G)

Since publishing *Atlas 0.5*, step **G** has become the primary focus of the Helix process. In implementation efforts conducted in 2016, the *Atlas* theory and proficiency model have been validated in a number of ways. During 2015, Helix began deploying *Atlas* with specific organizations in an attempt to use *Atlas* to establish the proficiency levels and career paths of participants and to be able to discuss ways to develop their careers in the future, towards achieving targeted levels of proficiencies required for particular senior positions within the organization (**G1**). In 2016, Helix helped a few organizations think critically about how *Atlas* could be implemented – including any modifications to fit the organizational context. (**G2**) The result of this work has helped the Helix team to clearly identify where tailoring of *Atlas* is expected versus where *Atlas* is expected to remain very consistent regardless of the organization, domain, etc. The primary expectations for tailoring are highlighted in the discussion of implications for use of the proficiencies (Section 4.4.9).

The first Helix workshop was held in July 2014, with participation of representatives from DoD, academia, and industry, including representatives from organizations that participated in Helix interviews. Feedback from the workshop significantly shaped *Atlas 0.25*. The second Helix workshop was held in August 2015 and reinforced the relevance and potential value of *Atlas* to a variety of systems engineering organizations. The third Helix workshop, an early adopter's workshop, was held in September 2016 and provided participants the opportunity to examine *Atlas* in detail and even allowed participants the opportunity to use the tools (**G3**).

2.4 QUALITATIVE ANALYSIS

The Helix research methodology discussed in the preceding sections was deployed using the analysis processes described below.

2.4.1 CODING

The interview dataset comprises nearly 6,000 pages of transcripts and summaries from 287 individuals. In order to make sense of such a large quantity of data, the Helix team uses qualitative data analysis, primarily through data coding. Coding is “a systematic way in which to condense extensive data sets into smaller analyzable units through the creation of categories and concepts derived from the data.” (Lockyer 2004) Codes can be layered, and evolve over time, as explained below. When developing a theory, as in the development of *Atlas*, categories and codes are generated *after* examining the collected data, aligning with the grounded theory approach. (Bourque 2004 and Lockyer 2004) The main type of coding done by the team so far is called “open coding”, the purpose of which is to break down, compare, and categorize data (Strauss and Corbin 2014).

The team has used two techniques for coding: auto coding and manual coding. "Auto coding" is only the first stage in parsing the information contained in transcripts and is not fully automated despite its name. Instead, as the team reviews and cleans each transcript, headings are added to the source documents to block out a large area of text as addressing a particular topic, such as personal characteristics, mentoring, experiences, etc. When the documents are imported into the qualitative analysis tool, NVIVO, the tool then automatically codes all text under that heading for the given subject. The team, then can pull up all auto coded text related to personal characteristics, for example, from across the entire data set and examine it at once. This allows a more consistent look at the related data that can then evolve more quickly, allowing the team to identify patterns that occur across data.

Auto coding is a useful approach, but has its drawbacks. One of the strengths of the coding approach is that codes can overlap - individuals may discuss several issues together and researchers can layer multiple codes together. Not only does this help to give a true characterization of the data, but common patterns in overlaps may provide useful insights. For example, the proficiency of big picture thinking was often discussed simultaneously with several of the values that systems engineers provide. This helped explain, for example, the relationship between big picture thinking as a critical skill and how that approach can provide value on diverse teams. But when using auto coding, layered codes are not possible; in the example of big picture thinking and value, the text would be tagged either as "Personal Characteristics" or "Proficiency" - not both. Since auto coding is only the first step, there are additional opportunities to create the layering and complexity that reflects the nature of the data. However, auto coding does limit the researcher to make a choice about what is most important or most prevalent in a section at the outset, which raises the risk that important relationships could be missed later. The other drawback to auto coding is that categories had to be developed and applied to all data and, therefore, could not happen early in the project. The team agreed to a limited set of categories, largely aligning with elements of *Atlas 0.5* published in 2015.

If auto coding was not used – for example, if there were a new area of inquiry, meaning that no headings had previously been identified and applied – then the team had to manually review and code all ~6,000 pages of data. Though keyword searches could be used, there was a risk that data could be missed if only keyword searches were used. For this reason, the team used a variety of keyword searches related to a given topic as well as a scanning read of a transcript when doing the initial pass for manual coding. For example, when looking for information on training, keywords included, "train," "course," "class," "learn", and "study". Once the initial coding was complete, this is essentially equivalent to auto coding in terms of level of depth.

Additional codes were then added to this subset of the data to further clarify the patterns. For example, a total of 30 individual personal characteristics were identified by participants. Some of these, in the discussion, were directly linked to the values that they helped to provide – these sections were double coded for both the characteristic and the value. Once all of the data had been analyzed, the team identified a reportable threshold – for example, for personal characteristics there were 141 excerpts and 30 characteristics.

Individual characteristics were mentioned anywhere from 15 times to only a single time across the excerpts. While none of the characteristics is "wrong", it was also not useful to simply provide a laundry list of items, particularly those that were only mentioned once across such a large dataset. It was more useful to first identify whether there were any relationships between items that might help identify areas of importance. This was done by comparing overlaps between codes. In other words, a single excerpt might be coded for multiple characteristics that were discussed together. By examining how often characteristics were cross-coded, it helped to identify relationships that participants believed are important across organizations. For example, in terms of personal characteristics, ambition and internal motivation often were discussed simultaneously, which is why they are grouped together in *Atlas*. Figure 3 provides an example of the coding comparisons conducted by the Helix teams. The higher the bars, the higher the overlap in coding between characteristics. This provides Helix with insight into relationships between and patterns around characteristics based on how interviewees discussed them.

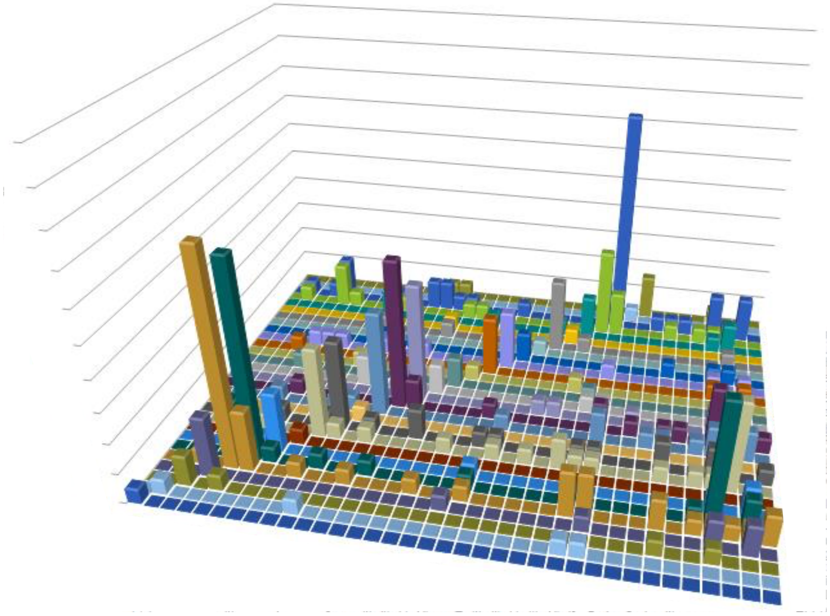


Figure 4. Example of Coding Relationships

When reporting, there was also a natural cutoff – again, in the example of Personal Characteristics, any characteristic mentioned in more than 6% of the excerpts. This was a natural threshold because the next cluster of characteristics occurred in approximately 1% or less. Items above the 6% threshold were reported in *Atlas 0.5*; items below it were not. However, because the coding structure remains in place, as additional data is added, if other characteristics become more prevalent, then they may be added to the next iteration of Atlas. Personal Characteristics were also presented in the descending order by the number of excerpts in the dataset; e.g., "self awareness" is listed first because it was discussed in the highest number of excerpts.

2.4.2 TOOL SUPPORT AND COMBINING QUALITATIVE ANALYSIS WITH DEMOGRAPHICS

To support the its analysis, the team has imported all of the data into NVIVO (QSR International 2016), a powerful and popular commercial qualitative analysis tool. NVIVO allows the team to code text as well as overlay additional information and identifiers about the sources. This replaces tools used early in the project, namely a combination of Dedoose (Dedoose 2016) and Microsoft Excel, which proved insufficient to handle the volume and diversity of data required. In NVIVO, the sources are tagged with a code for the organization in which the individual worked at the time of the Helix interview, and then each organization is linked to characteristics such as whether it is government or commercial; whether defense, healthcare, transportation, etc.; and how systems engineer are organized, e.g. embedded, matrixed, etc. As much as possible, each comment is also tagged for the individual who made it, though in the summaries from the earliest interviews, there are a few exceptions. Each individual, again, has several characteristics, including gender, whether they are a systems engineer or a peer, and some results of the Helix career path analysis.

2.5 CAREER PATH METHODOLOGY

In addition to the analysis of interview data, the Helix team developed a method for analyzing and visualizing the career paths of systems engineers. The career path method presented here supplements the qualitative data analysis described earlier with more quantitative information about an individual's career. This analysis was conducted for 181 systems engineers[1] from a dozen organizations. The initial data collection for career analysis was conducted by:

- Reviewing the resumes submitted by each individual, including chronology, organizations, position titles, and all descriptive text provided within the resumes;
- Reviewing interview transcripts and notes to add detail to the resume data;
- Reviewing the preliminary results during follow up interviews to clarify analysis. Individuals self-selected whether or not they would like to participate in follow-up interviews; roughly half of the individuals in the career analysis sample have participated in follow-up interviews; and
- Comparing the career paths with existing Helix research on the proficiencies of systems engineers and how career path elements may relate to these proficiencies. (Pyster et al. 2014b)

Using this approach, the Helix team developed a method to examine experiences and a common framework to capture, analyze, and visualize career paths. The self-assessment tool(s) provided to individuals participating in Helix to create their own career paths, including their proficiencies over time, are available in Appendix A.

2.5.1 CHARACTERIZING A SYSTEMS ENGINEER'S EXPERIENCES

Experimental literature on experiences has primarily focused on two metrics for experience: time (e.g. Ford et al. 1993; Schmidt et al. 1986; Firth 1979; Davidz 2006) and the frequency of times a specific task or activity of interest was performed (e.g. Stuart and Abetti 2002). Additional literature classifies human subjects based on their experiences – which is subtly different than classifying the experiences themselves – often using a combination of time and the frequency of tasks performed. This approach may also include considerations for specific roles played (e.g. Stuart and Abetti 2002, Kor 2003, Kirschenbaum 1992, Broome et al. 2001). Additional literature in the field of systems engineering, such as Sheard's "Twelve Systems Engineering Roles" (1996) or the *Graduate Reference Curriculum for Systems Engineering (GRCSE)* (Pyster et al. 2012) indicate, though, that the characterization of experiences is critically important to understanding how experiences enable growth.

The first challenge was to determine a common "unit of measure" for experience. Though time is common, it was not easily used in the data available. For example, if someone described a position they held over a five-year period, they did not explain the portion of time taken up by the activities they performed over those five years. In addition, several individuals submitted information on their careers that included detailed descriptions, but did not include markers for chronological time. Because of these data limitations, the Helix team chose to use a **position** as the unit of measure for experience.

Based on both the literature and the Helix data itself, each position has several characteristics:

- **Relevance:** A 'relevant' position is one that enables a systems engineer to develop the

---

[1] The interviews for all systems engineers in the Helix study are included here. However, the resume data was not provided for some of these individuals, and not all resume data was sufficient to complete each type of analysis. In general, the career analysis sample is N=157. Where the analysis looks at a subset of the sample or where individuals were eliminated from analysis for insufficient data, the sample size (N=x) is provided in the text.

proficiencies critical to systems engineering.

- **Position:** Every systems engineer who is employed at an organization fills a position that is established by the organization; that organization also defines the roles and responsibilities to be performed. Helix considers position as a 'unit of measure' for experience, since most of the characteristics of experience are in the context of the position that is held. A 'systems engineering' position is one where the individual's primary focus was on systems engineering activities.
- **Chronological Time:** The amount of time spent in any particular position or in performing a role.
- **Number of Organizations:** The number of different organizations that an individual has worked at, not counting internal movement within an organization across departments or divisions, reflects the variety of types of experiences that one may possess.
- **Organizational Sectors:** There are many differences in the general characteristics of an organization based on its sector. In *Atlas*, three organizational sectors are identified: government, industry, and academia.
- **Roles:** A role is a collection of related systems engineering activities. Roles were identified based on the activities consistently performed by systems engineers. There are 16 roles identified in *Atlas*, as described in Section 3.5, below.
- **Lifecycle Phases:** Generic systems engineering lifecycle phases considered in *Atlas* are based on the lifecycle phases in the *Guide to the Systems Engineering Body of Knowledge (SEBoK*). (BKCASE Authors 2015)
- **Systems:** There are many aspects to the types of systems on which a systems engineer could work. Working across these different categories provides valuable experience to an individual systems engineer.
    - **Domain:** This is the primary area of application for the systems being worked on. However, there are many domain categorizations; some domains also relate to industry sectors.
    - **Type:** Product systems, service systems, and enterprise systems are three major types of systems, depending on the nature and composition of the system of interest. System of systems is another paradigm in systems engineering, and could be a combination of one or more types of systems.
    - **Level:** A systems engineer could work on various levels of a system: component/element, subsystem, system, and platform or system of systems.

By using the data available for each individual, the characteristics of each position played and the order that they played them can be identified. Looking for patterns across the Helix data set, this information can be used to develop a preliminary understanding of how career paths shape proficiency.

The ways in which positions were categorized were pulled from existing literature wherever possible. For example, a systems engineer working in the commercial sector of a company may define life cycle in different terms than those used by a US Department of Defense systems engineer. To normalize the discussion, the definition of life cycle stages from the *Guide to the Systems Engineering Body of Knowledge* (SEBoK) was used; the interviewee's own words and phrasing were compared with the descriptions of life cycle stages in the SEBoK and categorized appropriately. (BKCASE Editorial Board, 2014) Likewise, the roles played by the interviewees were based on Sarah Sheard's "Twelve Roles of Systems Engineers" (Sheard 1996), although roles have been added to reflect what was seen in the data. Where existing literature was not available, categories were created that reflect the character of the data.

2.5.2 CHARACTERIZING A SYSTEMS ENGINEER'S EDUCATION

Education plays two key roles in the development of systems engineers. First, it provides the foundation knowledge to support engineering-related work. Typically, this takes the form of undergraduate education in an engineering discipline, technical field, or physical science. Second, graduate level education is an avenue to develop more advanced skills, explore more in-depth knowledge, and help systems engineers grow as they move through their careers.

The characterization of education was much more straightforward than the characterization of experiences. For each systems engineer in the sample, the team recorded:

- **Chronological Time:** The date of the completion of the degree program.
- **Type of Degree:** This is the level of education an individual achieved. The categories used were: bachelor's, master's, and doctor of philosophy (PhD). For this analysis, only education that resulted in a degree was recorded. Individuals did receive graduate certificates or took individual courses, but there was not enough data to draw any meaningful conclusions. Also, if a degree was in progress but not completed, it was not recorded.
- **Field of Study:** The primary discipline on which the individual's education was focused. These were initially recorded as reported. Over time, categories of related fields of study were created.

All systems engineers in the Helix sample held at least a bachelor's degree and the majority – 58% – held at least a master's degree.

2.5.3 IDENTIFYING KEY POSITIONS

A third aspect of career paths are the key milestones for a systems engineer's career. The Helix team focused on major steps or changes in a systems engineer's positions. A ***position*** is equivalent to the roles and responsibilities associated with an individual's title. Organizations will define what roles and responsibilities each position contains and *position* descriptions may not translate across organizations. The key positions identified for systems engineer included:

- **First systems engineering position.** This was self-identified by participants as the first position in which systems engineering responsibilities were the *primary focus* of a position, though they may have non-systems engineering responsibilities as well. This was often difficult to identify, because participants indicated that their roles often transitioned gradually and it was hard to identify when they officially became systems engineers, especially because so many never had that specific title. The Helix team recorded this information in whatever way it was provided by participants. In a few organizations, the hierarchy and structure for becoming a systems engineer was much more well-defined, and for individuals in those organizations, the transition to systems engineer was more easily identified.
- **Chief systems engineering positions.** A chief systems engineer (CSE) is someone who has formal responsibility to oversee and shepherd the technical correctness of a system, often coordinating with many other systems engineers who have smaller scopes of responsibility. These milestones are any positions in which an individual acted as a CSE, *regardless of their title within their organization*.
- **Project manager positions.** A project manager is someone who has formal responsibility to oversee the programmatic aspects of a system, generally focused on budget and schedule. Project

management responsibilities sometimes overlap with SE responsibilities, particularly those around planning and management; in some instances, a CSE may also function as a PM.

These milestones are important for understanding how the nature of a systems engineer's work has changed over time. It also gives insights into how quickly an individual progresses through different stages of her career. By comparing these patterns across individuals, common ranges of progression can be identified, as can outliers. For example, among the CSEs discussed in this paper, one individual became a CSE only 8 years after completing his undergraduate degree. However, only 12% of CSEs gained their first CSE position within 10 years after entering the workforce; therefore, this is an outlier rather than typical for for CSEs.

2.5.4 ASSESSING PROFICIENCY

Interviewees were asked about not only their common activities but what they believe were the critical knowledge, skills, abilities, behaviors, and patterns of thought (cognitions) that enable them to be effective in performing those activities. Helix calls these **proficiencies**.

By coding all of these responses individually and then aggregating like responses, the Helix team has identified the key proficiencies of systems engineers. These are elaborated in (Pyster et al. 2015). In brief, there are six proficiency areas, each of which contains several related groups of skills, or **categories**, as described below:

1. **Math/Science/General Engineering:** Foundational concepts from mathematics, physical sciences, and general engineering. Categories include: Natural Science Foundations; Engineering Fundamentals; Probability and Statistics; Calculus & Analytical Geometry; and Computing Fundamentals.
2. **System's Domain & Operational Context:** Relevant domains, disciplines, and technologies for a given system and its operation. Categories include: relevant domains, relevant technologies and systems; relevant disciplines; familiarity with the system's concept of operations.
3. **Systems Engineering Discipline:** Foundation of systems science and systems engineering knowledge. Categories include: lifecycle; systems engineering management; systems engineering methods, processes, and tools; and system complexity.
4. **Systems Engineering Mindset:** Skills, behaviors, and cognition associated with being a systems engineer. Categories include: big-picture thinking; paradoxical mindset; flexible comfort zone; abstraction; and foresight and vision.
5. **Interpersonal Skills:** Skills and behaviors associated with the ability to work effectively in a team environment and to coordinate across the problem domain and solution domain. Categories include: communication; listening and comprehension; working in a team; influence, persuasion, and negotiation; and building a social network.
6. **Technical Leadership:** Skills and behaviors associated with the ability to guide a diverse team of experts toward a specific technical goal. Categories include: building and orchestrating a diverse team; balanced decision making and rational risk taking; managing stakeholders and their needs; conflict resolution and barrier breaking; and business and project management skills.

In addition, the non-systems engineers in the sample – project managers, classic engineers, executive leadership, and human resources personnel (HR) – were asked which proficiencies they considered critical in the most effective systems engineers with whom they worked. These were also coded and aggregated with the systems engineers' responses; they validated the existing categories.

In 2015, the Helix team provided the and reviewed the draft proficiency model to participants and had them react to the categories and structure directly. The existing structure was validated, with no additional skills being cited that did not fit within existing categories; this did, however, help the team in re-allocating some proficiencies to other categories to make them more easily understood by a wider audience.

Finally, systems engineers were asked to perform self-assessments of their own proficiencies at different points in their careers, which could then be overlaid with their career paths. Early on, the Helix team would perform its own assessments during these discussions and map them against the self-assessments to ensure alignment between the team's approach and the participants'. They were also asked to cite what they believed were the most critical proficiencies for their *current* positions. In addition, some were asked to identify what they believed were the minimum proficiencies to be effective in their current positions. Non-systems engineers also did self-assessments, to help identify where these proficiencies overlap with other disciplines. In addition, they were asked what they believed were the most critical proficiencies or the minimum proficiency level they would desire in the systems engineers that they work with. All of this work helped to validate the proficiency set as a useful and comprehensive model.

The forces identified in Figure 1 – experiences, mentoring, education and training – are linked to the growth of proficiency by interview data. When an individual would cite a critical skill, the Helix team would ask how that individual had developed that skill over time. These types of discussions were cross-coded for both the relevant force(s) and the related proficiency(ies). Again, the self-assessment tool for proficiency can be found in Appendix A.

2.5.5 MAPPING A CAREER TIMELINE

As described above, chronological time is an attribute of all positions. The final step in developing a career path map for an individual was to create a visualization over time of all of the elements listed above. This visualization lays out all of an individual's positions, and their characteristics over time, with their education, the career milestones, and their proficiency assessments. Figure 4, below, shows a generic example of this.

In Figure 4, only the timing, roles and the lifecycle stages characteristics of positions are illustrated. This is for two reasons: one is ease of visualization in a single graphic – though any combination of attributes is possible in this format – and the second is that systems engineers were able to provide the clearest discussions on how *roles* and *lifecycle* exposure contribute to proficiency. For other attributes, these relationships were more sporadically represented in the data; in addition, not all systems engineers provided basic data on all attributes, but the Helix team was able to complete roles and lifecycle data for nearly the entire dataset (93% for roles; 91% for lifecycles).

By creating these individuals "maps" for each career path, it is possible to start identifying patterns – not only in proficiencies but in the common attributes that lead to similar proficiency profiles. Additional analysis of the career paths of individuals in similar roles was also insightful; even though there is some individuality to each systems engineer's career, the common patterns indicate ways that systems engineers may typically grow – or areas where certain types of systems engineers differ from others. The analysis highlighted in this paper is that of chief systems engineers – not only of their career paths overall, but in a few critical cases, highlighting their differences from other senior systems engineers.

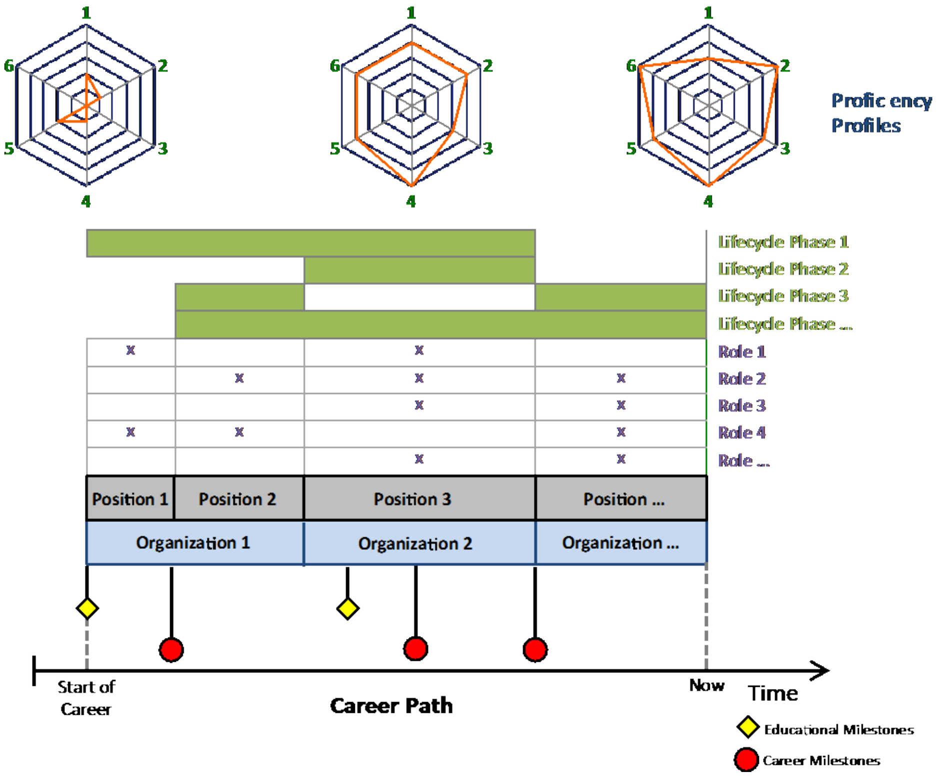


Figure 5. Generic Career Path Mapping

2.5.6 Steps to Validate Findings from Career Paths

There were several ways in which the findings from career paths were verified and validated. The most straightforward was verification through follow-up interviews, where individuals were presented with the current analysis of their career paths and asked to provide any corrections or fill any gaps. Though only 48% of interviewees also participated in follow-up interviews, the changes and updates were minimal, and often reflected additions where certain aspects of an individual's career had not been discussed in the initial interviews. Because the reviews by individuals of their own career paths revealed no major issues with the methods, the Helix team considered that the method is a valid approach to understanding the causes of change and growth over time.

In terms of additional validation, the Helix team acknowledges that, for a number of reasons, the way an individual progressed through her career may not have been an optimal approach. Participants were asked to identify areas where what they would have approached their careers differently based on their current levels of insight. They were also asked their general satisfaction with their career progression and to identify areas where colleagues might have had a different, preferred approach. And because – as explained in the dataset discussion above – the interviewees identified themselves and their organizations identified them as all of "average" to "excellent" effectiveness, it is reasonable to draw conclusions about the career paths of these individuals.

Finally, in 2015, the Helix team worked with one organization to pilot the career path approach. Individuals worked through an example career path with the Helix team and then mapped their own careers in real time. The feedback was that this approach was much better structured and focused than any career guidance they had received. In some cases, this reinforced that their current planning was appropriate, and other individuals reported that with insights from their career paths, they realized that they needed to seek new opportunities. All of the participants (n=34) agreed that this approach was useful and would be a valuable tool for them as individual systems engineers and for the organization.

Details on the *Atlas* proficiency model can be found in Section 4.4.

2.6 HELIX DATA

Helix research uses a bottom-up approach, based on the data being analyzed. Hence, it is essential to gather data that is sufficient in quantity and quality to enable effective development of *Atlas*, and to provide reliable insights and recommendations that can be confidently used for the development of effective systems engineers.

2.6.1 DATA SOURCES

The primary source of data for Helix research is face-to-face semi-structured interviews with participants at their place of work. Additional information about the participant and the organization were also collected as available. Another data source that Helix gained access to was the application data for the INCOSE Systems Engineering Professional (SEP) certification program.

Helix Interview Data

From June 2013, when Helix conducted its first site visit for data collection, until November 2015, a total of 289 participants were interviewed from 21 organizations. Typically, 2 to 3 members of the Helix team interviewed anywhere from 1 to 6 participants in a single interview session.

Interview participants, if willing, also provided their resumes with details about their educational background, work experiences, and any other information they wished to provide.

Follow-up interviews were conducted over telephone with willing participants, to explore topics that could not be covered in the initial face-to-face interviews or to collect additional information based on their resumes. Follow-up interviews were also used to validate results of Helix analysis.

In both the initial interviews as well as follow-up interviews, transcripts were created when audio recording was permitted; when not permitted, summaries were prepared from notes taken during the interviews. These transcripts and summaries from a total of about 270 hours of interviews form the bulk of data that Helix analyzed.

The data that was analyzed for *Atlas 0.5* and presented in this report is limited to a sub-set of interviewees from DoD and DIB organizations. Subsequent reports will include additional analysis performed on the Helix interview data.

INCOSE SEP Application Data

INCOSE provides three different levels of SEP certification: Associate (ASEP), Certified (CSEP), and Expert (ESEP). Applicants from all over the world seeking INCOSE certification apply for the appropriate level based on their systems engineering experiences, knowledge, and accomplishment. INCOSE provided to Helix, under a Non-Disclosure Agreement, over 3000 application forms received from applicants during the period May 2004 to May 2014. Though the application data was available in electronic form, it was not in a format that would readily support analysis. Significant time and effort was spent in extraction, cleaning, and tabulating the data to enable further analysis.

Analysis of INCOSE data did not directly contribute to the building of *Atlas*, but provided some validation and additional insights for the analysis of the interview data.

2.6.2 DEMOGRAPHICS OF SAMPLE POPULATION

Understanding the sample population is important, since the interview data is reflective of the population from which it has been collected, and in turn, that data is the basis for *Atlas*. An understanding of the INCOSE applicants reveals the breadth of the data that it contains.

Demographics of Interview Population

Among the 288 participants across all Helix interviews, 78% were male and 22% were female. For the 107 individuals whose interviews are in the expanded dataset, that ratio was 71% male and 29% female.

The Helix team created a detailed rubric for understanding the seniority of systems engineers. That rubric was validated with systems engineers' normalized self-assessments of their own seniority. (Pyster et al. 2015) The team did not have sufficient data to create seniority rubrics for the individuals in each of the other disciplines reported here. Therefore, the team simply recorded the self-assessments of these individuals. A comparison of the seniority profiles of the systems engineers in the original and expanded datasets can be found in Figure 6.

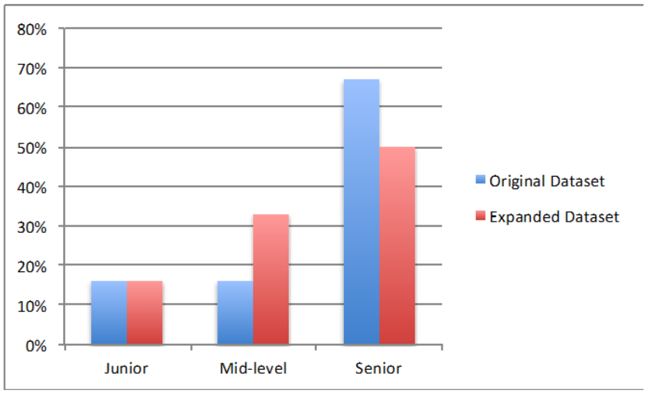


Figure 6. Seniority of Helix Interview Population

Among the 20 organizations that have participated in Helix interviews, 11 were from the DoD or DIB; the other 9 were from healthcare, transportation, IT, and telecommunications. Note that this does not mean that the organization focused *only* on these areas, but rather the segment of the organization that participated in Helix was focused on these areas. Figure 7 provides an overview of the percentage of individuals from each of these sectors or domains in the expanded data (of the 107 total). In Figure 7, blue represents the original dataset analyzed for the creation of *Atlas 0.5* and red represents the dataset for this expansion.

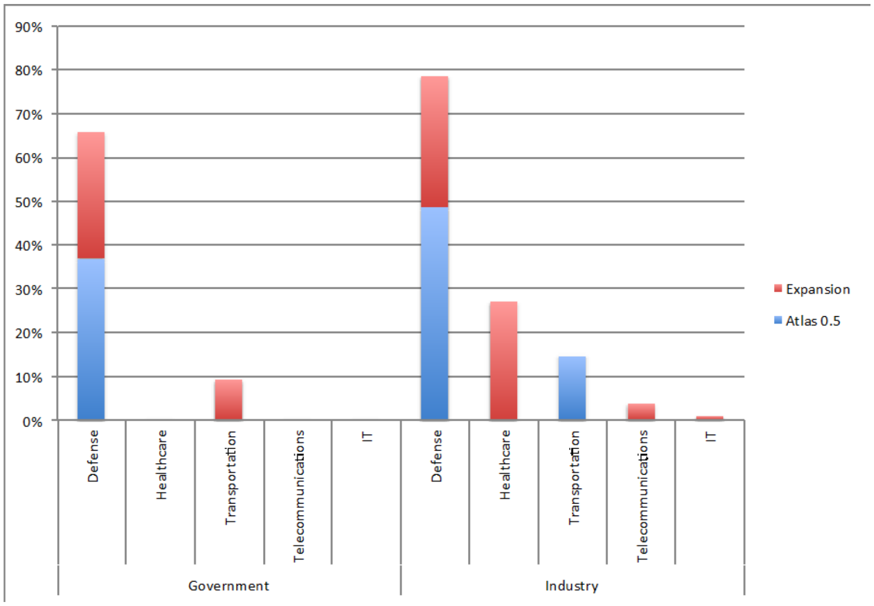


Figure 7. Breakdown of Current Sample (N=107) by Domain across Government and Industry

In terms of the types of positions filled by these individuals, the analysis was broken down according to discipline: systems engineering (SE), program/project management (PM), and classic engineering disciplines including electrical (EE), mechanical (ME), software (SwE), and "other" engineering. Figure 8 provides an overview of the breakdown of the sample along these lines (N=107). Of the expanded interviewees, 77% were systems engineers and 23% were non-systems engineers. Over half of the non-systems engineers in the expansion were project managers and 45% were classic engineers. The "other" engineers in the sample are engineers focused on a specific technology such as radar or ultrasound.

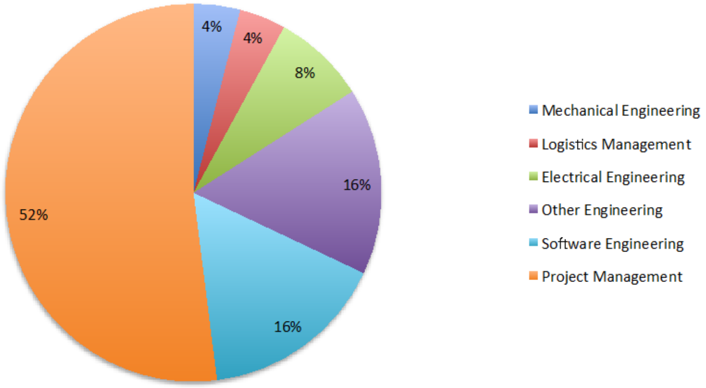


Figure 8. Disciplines of Non-Systems Engineers in The Expanded Sample

The percentages in Figure 8 need some interpretation. At times, individuals were identified as belonging to a discipline by their organizations. However, when they were asked whether or not their positions were in systems engineering, many gave responses such as, "well, I do a mix of things, but I definitely perform some systems engineering activities." The individuals reflected in Figure 8 (23% of the expanded dataset) *self-identified* as non-systems engineers. There were other individuals identified as non-systems engineers by their organizations who stated in their interviews that although their titles reflect a different position – such as mechanical engineering – their primary responsibilities really fell more under systems engineering, and therefore, they self-identified as systems engineers.

1.1. INTERPRETATION AND GENERALIZATION USING THE DATASET

Helix is careful when using the data to understand whether and how findings and conclusions about the dataset can be generalized to the wider population of systems engineers. The team recognizes that there are some limitations based on the sample. Though it is relatively large at nearly 300 individuals, there is no clear estimate of how many systems engineers are working in the US, let alone the world; this makes it difficult to understand how statistically representative the sample may be. Likewise, all interviews conducted to date have been in the US; though a few individuals provided insight into, for example, education outside the US, and some organizations had units outside the US, current findings reflect a US context. Likewise, though Helix has expanded beyond its initial defense roots to include healthcare, transportation, telecommunications, and other industries, Figure 3 clearly shows that the majority of individuals participating in the sample are from the defense industry. Given these limitations, the Helix team is careful not to over-interpret the data.

However, even with the limitations of the sample, the team believes the overall size of the sample and the diversity in terms of industries, organization types, and seniority makes any findings and conclusions drawn from the data extremely useful, even if they are not fully generalizable. The Helix team first published *Atlas* 0.25 in 2014; since then the coverage of industries and organizations has grown and the team has nearly doubled the size of the sample. And over this time, there have been some updates and edits, but no major issues or breaks with the theory have been discovered. This, again, builds confidence that the existing sample is sufficient to enable useful and insightful work. As the team continues work on implementation and application of *Atlas* (see Future Directions), these activities should generate greater confidence in the generalizability of the data.

2.6.3 DEMOGRAPHICS OF INCOSE SEP APPLICANTS

From the over 3000 application forms, about 2500 unique applicants were identified for further analysis. These applicants were predominantly from the U.S, but there were others from Asia and Europe as well, as indicated in Table 1.

Table 1. Geographical Distribution of INCOSE SEP Applicants

Rank	Country	# of Applicants	% of Total
1.	U.S.	1847	74%
2.	India	179	7%
3.	Germany	151	6%
4.	France	101	4%
5.	U.K.	49	2%
6.	Sweden	41	<2%
7.	Spain	36	1%
	Other	100	4%

Information from all the 2504 unique applicants was used for analysis of education background; a subset of those applicants was analyzed for experiences.

3 INTRODUCTION TO *ATLAS*

According to Merriam-Webster, a theory is a set of general principles or ideas relating to a particular subject (2014). *Atlas* is a set of general principles and ideas that relates to the subject of what makes systems engineers effective and why. In doing so, *Atlas* also provides insights into how individuals can develop into effective systems engineers throughout their careers and what organizations can do to support this development.

3.1 ATLAS OVERVIEW

The overview of *Atlas* in the context of an individual systems engineer employed in an organization is captured in the systemigram illustrated in Figure 9. A systemigram consists of nodes that contain noun phrases, links that contain verb phrases, and is to be read as sentences along the direction of the arrows. The primary sentence is read from the top left node to the bottom right node and presents the main theme of the systemigram. In the ensuing discussions, sentences to be read in the systemigram are italicized, where nodes are represented in square brackets.

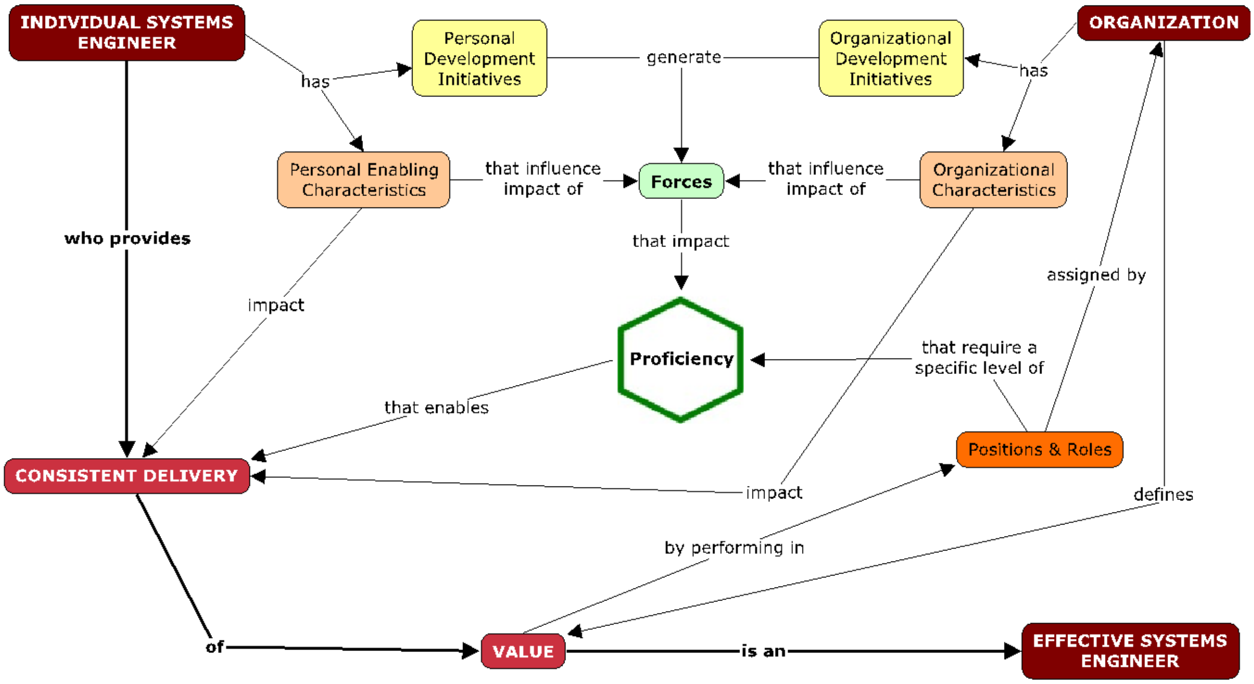


Figure 9. *Atlas 1.0*

From Figure 9 above, it can be seen that the main theme of *Atlas* is: '*[Individual Systems Engineer] who provides [Consistent Delivery] of [Value] is an [Effective Systems Engineer]*'. This fundamental definition of an effective systems engineer hinges on *[Value]*, and it can be seen that '*[Organization] defines [Value]*'. Therefore, it is on the organization to define the value that the systems engineer is expected to provide. Further, the individual systems engineer provides '*[Value] by performing in [Positions and Roles] assigned by [Organization]*'. Therefore, it is again on the organization to establish the position of the systems engineer in terms of roles and responsibilities, keeping in mind that '*[Positions and Roles] require a specific level of [Proficiency] that enables [Consistent Delivery] of [Value]*'.

The core of *Atlas* is the proficiency of the individual systems engineer – what proficiency means, and how it can be improved. '*[Individual Systems Engineer] has [Personal Development Initiatives]*' and '*[Organization] has [Organizational Development Initiatives]*'; together, they '*generate [Forces] that impact [Proficiency]*'. At the same time, '*[Individual Systems Engineer] has [Personal Characteristics] that influence the impact of [Forces]' and '[Organization] has [Organizational Characteristics] that influence the impact of [Forces]'* – these forces may have a positive or a negative influence. Further, both personal enabling characteristics and organizational characteristics '*impact [Consistent Delivery] of [Value]*'; again, the impact can be positive or negative. Amidst all these influences and impacts, the challenge for the individual systems engineer and the organization is to improve the '*[Proficiency] that enables [Consistent Delivery] of [Value]*' to the organization.

The color-coding of the systemigram (Figure 1) is designed to show the relationships between various elements of *Atlas* as follows:

- The mainstay – the primary focus of the systemigram (Boardman and Sauser (2008) – is in red.
- Primary actors are in dark red (individual systems engineer and organization, leading to the desired end state of effective systems engineer).
- Elements related to the skills of systems engineers – the specific skills themselves or how they are developed – are in green.
- Characteristics of the primary actors are in orange.
    - "Positions and Roles" is called out as a darker orange because it is related to both the individual and the organization. Positions and roles are a characteristic of the current state of the individual and the mechanism by which she would deliver value. These are also determined by the organization, and the way an organization defines positions is related to other organizational characteristics.
- Initiatives of the primary actors are in yellow.

3.2 DYNAMIC ASPECTS OF ATLAS

The *Atlas* overview illustrated in Figure 9 can be considered as a quasi-static snapshot in time, but many of the elements of *Atlas* are dynamic in nature. The level of proficiency of an individual systems engineer is not fixed, but is constantly changing due to the impact of forces over time. Similarly, other elements of *Atlas*, including characteristics and initiatives of the individual systems engineer and of the organization, continue to change over time. Further, as the level of proficiency of an individual systems engineer increases over time, the organization is likely to place that systems engineer into different positions.

This dynamic aspect of *Atlas* is not captured in the overview, but is reflected in the career paths of individuals over time, where an individual's career path is the precise combination of the forces they undergo in the positions and roles they perform in over their entire career.

Leading up to the publication of *Atlas 1.0*, the Helix team defined methods to depict and analyze the career paths of systems engineers and used those methods to analyze the systems engineers in its interview sample, and how those systems engineers are shaped by the impact of forces and positions and roles over time.

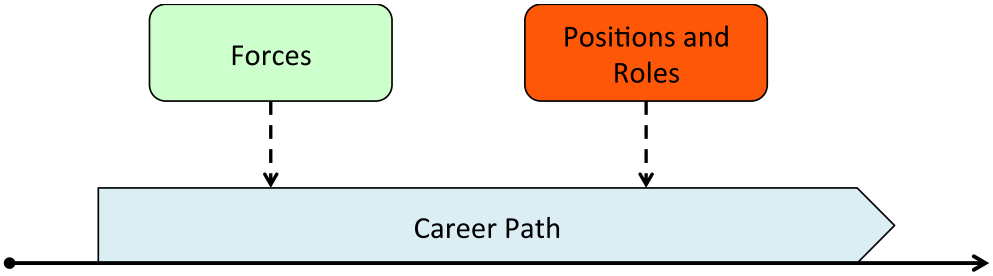


Figure 10. Career Path: A Dynamic View of *Atlas*

The Helix team has defined methods to depict and analyze the career paths of systems engineers. The team used those methods to analyze the systems engineers in its interview sample and to understand how those systems engineers are shaped by the impact of forces and positions & roles over time.

3.3 ATLAS DEPLOYMENT

The intent is to develop *Atlas* so that it can be independently deployed by individuals and organizations. Since *Atlas 0.25* was mature enough for trial, it was used in some organizations during 2015 to assess the proficiency levels and understand the career paths of individual systems engineers. Likewise, *Atlas 0*.5 was used in 2016 to validate the findings and understand the tailoring required to implement *Atlas* in practice. Feedback and observations from these trials have influenced the development of *Atlas 1.0*, as described in this report. Approaches for implementing *Atlas* are reflected in Section 6 of this report.

4 ATLAS: THE THEORY OF EFFECTIVE SYSTEMS ENGINEERS

This Chapter provides an overview of *Atlas* and frameworks and various elements of *Atlas*. Section 6.1 presents underlying definitions and classifications that are fundamental to *Atlas* and to all subsequent discussions across this report. Sections 6.2 through 6.7 elaborate on the different elements of *Atlas*, as indicated in Figure 11 below.

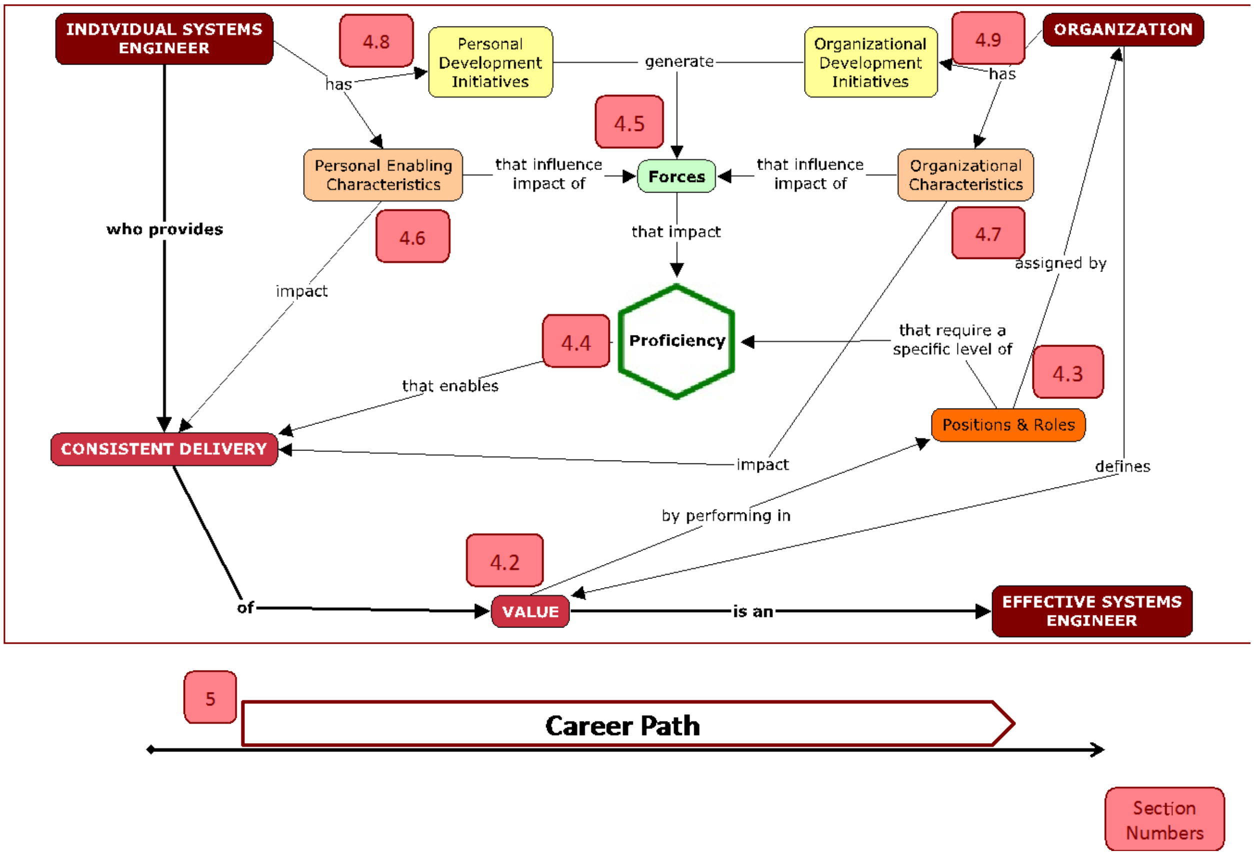


Figure 11. Elements of *Atlas* and Coverage in This Report

Atlas is the product of the research and analysis performed by Helix. Since a grounded theory based approach was followed, *Atlas* is largely the aggregation of the themes and constructs identified from the data.

4.1 ATLAS CLASSIFICATIONS

- **Seniority of a Systems Engineer**

  As systems engineers traverse the path of their careers from the point of entry into the workforce (or recruitment) to the point or exit from the workforce (or retirement), there is a continual maturation that is reflected in the breadth and depth of their proficiencies; the types of roles & positions they play; and the value that they provide or that is expected from them. Grouping systems engineers under some levels of 'seniority' that reflect the levels of maturation enables patterns to be identified across systems engineers, and insights to be drawn from them.

  Helix has identified three levels of seniority in systems engineers: junior, mid-level, and senior. Traditionally, 'number of years of work experience' has been used as a preliminary criterion for distinguishing between these levels of seniority, but it fails to capture the nuances of differentiation within systems engineers. Hence, it is not included in Table 2 that states various criteria used to distinguish between junior, mid-level, and senior systems engineers. These criteria are meant to be indicative and not rigid; there are always examples of specific individuals whose seniority is not consistent with these criteria.

Table 2. Career Criteria for Distinguishing the Seniority of Systems Engineers

	Junior	Mid-level	Senior
1.	Not more than 1 formal leadership position	At least 2 formal leadership positions	More than 2 formal leadership positions
2.	Experiences primarily in components	Experiences in components and subsystems, and perhaps in systems	Experiences in components, subsystems, systems, and perhaps in systems of systems
3.	Experiences in at least 2 aspects of the systems lifecycle	Experiences in at least 3 aspects of the systems lifecycle	Experiences in at least 4 aspects of the systems lifecycle

With respect to Table 2:

1. Experience is considered to be 'relevant' if it directly supports the growth of systems engineering proficiencies.
2. A leadership position is 'formal' if it is officially defined and recognized by the organization. This does not mean that the individual necessarily has organizational authority over the individuals she is leading. Likewise, there is no defined minimal team size. Typically, early leadership positions are over small teams (less than five people) and as the individual matures, the size of the teams increases.
3. The hierarchy of system levels (components -> subsystems -> systems -> system of systems) is based on definitions from the *Guide to the Systems Engineering Body of Knowledge* (BKCASE Editorial Board 2016) and reflects system complexity and completeness, where 'parts' at any level are combined to form the 'whole' at the next level.
4. The various aspects of the systems lifecycle are based on definitions from the *Guide to the Systems Engineering Body of Knowledge* (BKCASE Editorial Board 2016) and are elaborated

in Section 4.4.5.

5. Formal education, titles, and roles are *not* considered to be distinguishing criteria, since they cannot be used to consistently draw any distinctions between levels of seniority of systems engineers. However, as a baseline, systems engineers typically have an undergraduate degree in a STEM (science, technology, engineering, and mathematics) field.

Seniority, as described in Table 2, is specific to what an individual has done. This is based upon the Helix data. In interviews, individuals were asked to self-assess their seniority level and provide their rationale. Nearly all of the individuals asked provided a rationale based upon what they had or had not done that qualified them for a particular level of seniority. Individuals found it very difficult to articulate an expectation around capabilities rather than their career paths. This is why Table 2 focuses on careers.

In a related SERC research task (RT-149) focused on leadership development, a parallel rubric for assessing seniority was developed based on capability rather than how these capabilities were developed. This is reflected in Table 3, below.

Table 3. Career Stages of Systems Engineers (from Felder et al. 2016)

Responsibility Type	Junior	Mid-level	Senior
People Responsibility	Managing one's self	Managing a team	Managing managers
Program Responsibility	Not responsible for any program	Decision-making authority over programs having a limited to moderate level of size, scope, and complexity	Decision-making authority over programs having a large level of size, scope, and complexity
Knowledge Responsibility	Introductory level of professional knowledge	Intermediate level of professional knowledge and expertise	Subject matter expert, expanding breadth and depth

Though there is not a 1:1 correlation between the elements of the Helix seniority framework and the RT-149 framework, they are clearly related. For example, Helix criteria 2 speaks to the level of system responsibility and individual has experienced; this relates to the "program responsibility" of RT-149 and could be considered a proxy for size, scope, and complexity. Likewise, experiences across the lifecycle of systems could be a proxy for knowledge responsibility – an individual who has experiences across the entire lifecycle may be more of a subject matter expert in systems than an individual who has experienced little of the lifecycle.

Table 2 represents current seniority classifications based on the Helix data. In 2017 the Helix team will use existing literature such as the results of RT-149 and other relevant frameworks to further refine the seniority criteria.

4.2 VALUE OF SYSTEMS ENGINEERS

The broad question that Helix is trying to address is: 'How to develop effective systems engineers?' The key term in this question, in addition to a consistent understanding of who is a systems engineer, is 'effective'. When initially asked who an 'effective systems engineer' was, interviewees tended to give the response '*one who develops (or supports development of) systems within time, cost, and schedule constraints*'. This definition was not found to be very insightful, and hence Helix defined an effective systems engineer as '*someone who consistently delivers value by performing systems engineering activities*'. This definition introduced the term 'value', and thus provided a context for the effectiveness. Of course, value by itself is a subjective term, and was not something that Helix wanted to define up front. Instead, Helix wanted to understand what systems engineers said was the value they provided and to understand what non-systems engineers said was the value that systems engineers provided.

The Helix team probed on the concept of value in 100% of the interviews conducted. The discussion of value took two general forms: an individual's perspective of the primary value that she provides as a systems engineer and an individual's perspective of the overarching value that systems engineers in her organization provide. Some individuals answered the value question in ways more readily linked with *proficiency* than value; for example, they might have referenced communication skills or deep understanding of the systems engineering processes. And as indicated above, a number of systems engineers also defined value in terms of overall project success ("on time, within budget"), which does not allow specific insights for systems engineers versus project managers or any other personnel who support the project. After filtering these types of responses, there were 313 individual excerpts on the value that systems engineers provide offered from 85 individual systems engineers.

The key values identified to date are provided in the list below. The main bullets state the overarching values that systems engineers provide; the sub-bullets are the ways these values are achieved, often discussed as enabling or lower-order values. Percentages reflect the percent of the data related to a given value or the relationship between values. So for example, the first value, "Keeping and maintaining the system vision", was described in 11% of the excerpts on value. However, in 39% of the areas where "Keeping and maintaining the system vision" was discussed, understanding of the customer's true requirements was described as a key enabling value. In some instances, percentages are not provided; these areas require additional analysis.

The primary values that systems engineers provide include:

- Keeping and maintaining the system vision (11%) is enabled by:
    - Getting the "true" requirements from the customer and creating alignment between the customer and the project team. (39%)
    - Seeing relationships between the disciplines and helping team members understand and respect those relationships. (33%)
    - Balancing technical risks and opportunities with the desired end result. (36%)
    - Providing the big picture perspective for the system. (44%)
- Enabling diverse teams to successfully develop systems. (10%)
    - Effectively understanding and communicating the system vision to the team, and ensuring that the team is aligned with this vision. (38%)
    - Helping the team to understand the big picture perspective and where they fit within the larger picture. (38%)

  - Identifying areas of concern for integration in advance. (13%)
- Managing emergence in both the project and the system (7%)
  - Projecting into the future (14%), which includes staying “above the noise” of day-to-day development issues and identifying pitfalls.
  - Technical problem solving balanced with the big picture perspective. (43%)
- Enabling good technical decisions at the system level (7%)
  - The ability to see the vision for the system and communicate that vision clearly is a key enabler to helping teams make good technical decisions. (40%)
  - The big picture perspective is critical for understanding the system holistically and enabling system-level technical decisions, versus decisions made at the component or sub-system level. (22%)
  - A systems engineer’s solid grasp on the customer’s needs is also a critical enabler to ensuring that decisions made will keep the system on the correct technical path. (22%)
  - Being able to bring together a diverse team of engineers and subject matter experts is also critically important. (26%)
  - A systems engineer’s problem solving abilities – particularly the ability to focus on root versus proximal cause – is also a key enabler. (26%).
- Supporting the business cases for systems (7%)
  - Balancing traditional project management concerns of cost and schedule with technical requirements. (41%)
  - Understanding the position of a system within the organization or customer’s portfolio and communicating this to the team. (59%)
- Translation of technical jargon into business or operational terms and vice versa (11%)
  - Translating highly technical information from subject matter experts into common language that other stakeholders can understand.
  - Translating operational concepts, customer needs, and customer desires into language that makes sense for engineers and program managers who do not have the same understanding of the systems’ future operating environment.

4.2.1 VALUES THAT SYSTEMS ENGINEERS PROVIDE OUTSIDE OF DOD

Individuals in the expanded sample were asked several questions to help validate the value provided by systems engineers. Systems engineers from outside the defense sector were asked the top value(s) that they personally provided as well as their perspective on the key value(s) provided by systems engineers overall. Project managers and classic engineers were asked, from their perspective, the key value(s) that the most effective systems engineers that they personally had worked with provided as well as their perspective on the value(s) that systems engineers *should* provide.

In particular, there is an appreciation for the big picture perspective systems engineers provide. This is particularly important for enabling successful integration activities, which were seen as another key value of systems engineers.

Systems engineers are also generally seen as effective at working with the customer; outside the defense sector, there is a generally stronger focus on customer experiences. Several interviewees who had previous experience in the government, but currently work in industry, explained that in a DoD environment, the Joint Capabilities Integration and Development System (JCIDS) process is often executed such that requirements are provided *to* systems engineers; often, they do not have an opportunity to participate in requirements development from the beginning of a project. Also, in the DoD environment, systems engineers are often separated from their external customers; so while a focus on the customer's needs *is* stated as important in the defense community, opportunities for direct interactions can be more limited.

In general, whether in the defense sector or other business sectors, systems engineers in government organizations tended to be more focused on providing value by emphasizing standard processes, while commercial organizations tended to focus more on delivering the "right" end results by asking good questions, generating a vision for the system, and providing the big picture perspective. This does *not* mean that systems engineers in government organizations value process over the end result of systems development; instead, it means that in an acquisition environment – which was the context for the majority of government systems engineers – following a rigorous process was seen as a primary way to provide value and help achieve end results. In commercial companies, process was discussed, but not seen as a key means for providing value, although systems engineers in commercial companies did state that systems engineers provide value by bringing a logical approach to problem solving. In some organizations, processes were seen as a way to institutionalize these types of approaches, although with varying degrees of success. It is worth noting that systems engineers in commercial organizations in highly regulated industries tended to emphasize process more strongly than their counterparts in less regulated commercial enterprises.

The values reported in *Atlas 0.5* are described in Section 3.2 above. The expanded data supports these findings. Any major differences seen in the expanded data set are described below. The values described in *Atlas 0.5* translated well outside of the defense community. This was the view not only of the systems engineers in the expanded dataset, but also of their peers.

There is a generally high respect for *effective* systems engineers across domains and organizations. Effective systems engineers, again, provide the values described in *Atlas*. However, program managers and classic engineers also stated that a problem with systems engineers, as a population, is that not all systems engineers *really* provide those values. When they do so, they are seen as critical to the success of programs, systems, and the organization. But in some organizations, their peers felt that up to a third of systems engineers are overly focused on rigid application of process, adding burden to programs rather than insight and value. This was consistent between government and industry organizations.

4.2.2 VALIDATION OF VALUES SYSTEMS ENGINEERS PROVIDE

Project managers and classic engineers were also asked what they felt were the top value(s) they provided through their own work, in addition to the insights in the values provided by systems engineers. Although these individuals in the expanded dataset were all mid-level or senior, they offered a few observations about more junior individuals in their disciplines:

- Classic engineers tend to provide value through the application of their education at a very detailed level. Electrical engineers may design a circuit or software engineers may write a section of code. It is their skills as individual contributors that are valued early in their careers.
- There are two main tracks for people in the classic disciplines based on their primary value:
    - Those highly valued as individual contributors will generally continue deeper into their disciplines, becoming subject matter experts within their particular domain.

    - Those who are successful as individual contributors, but also provide a broader perspective – for example, asking how changes might impact other parts of the system or trying to understand more clearly how customers intend to use a system – tend to become leaders within their particular domain.
    - Both the deep specialists and the broader experts were seen as critical to the successful development of systems, but their positions must be tailored to the types of values they more naturally provide. A mix of the two ends of the spectrum is critical for technological innovation and advancement.

Program/project management is not generally suited to junior-level individuals. At the mid- and senior levels, the most common themes heard were ways to help a program stay on track with planned budget and schedule. However, there were also overlaps with some of the values of systems engineers, although again often from a budget, cost, and resources point of view rather than a technical one:

- Enabling diverse teams to successfully develop systems.
    - Helping the team to understand the big picture perspective and where they fit within the larger picture.
- Managing emergence in both the project and the system
    - Projecting into the future, which includes staying "above the noise" of day-to-day development issues and identifying pitfalls.
    - Technical problem solving balanced with the big picture perspective.
- Supporting the business cases for systems
    - Balancing traditional project management concerns of cost and schedule with technical requirements.
    - Understanding the position of a system within the organization or customer's portfolio and communicating this to the team.

As described above, in the expanded dataset, there is clearly overlap between the values provided by program manager and leaders in classic engineering disciplines. Some of this overlap is complementary – for example when "projecting into the future", program managers tend to focus on things that might impact cost and schedule more than technical issues; systems engineers focus on technical impacts; together, there is a much better and more holistic understanding of the overall potential risks to a system.

4.2.3 IMPLICATIONS FOR USE

The values that systems engineers provide, outlined in *Atlas,* provide a good starting point for organizations to define their end-state expectations for systems engineers' effectiveness. This does not mean that every organization will expect every value from each systems engineer – and likewise there may be expectations from the organization that are not captured in *Atlas*. However, using this list and adding or editing as appropriate for the organizational context can help to set very clear expectations about what a successful systems engineer delivers. This information can then be communicated in a variety of ways: individuals will understand the goals and expectations more clearly; managers will have a clearer basis on which they are making personnel selections; and leadership will develop additional language to clarify who systems engineers are and what the expected benefits to systems engineers' participation in teams will be.

4.3 ROLES AND POSITIONS

An individual systems engineer fills a *position* (or holds a title) in an organization, and there are many *roles* that the systems engineer is expected to perform in that position. *Atlas* identifies 17 systems engineering roles; typically, a systems engineer performs a combination of these roles while holding a single position. Starting with the 'twelve systems engineering roles' identified by Sheard (1996). The Helix team recombined, renamed, removed, and added roles to reflect the Helix data collected during interviews about the activities systems engineers perform in organizations today. This was socialized with the community through conference papers and presentations, the Helix workshops, and through early adopter activities with several organizations.

Table 4, Table 5, and Table 6 provide the roles of systems engineers and provides an explanation of how each role came to exist in the framework. For example, "System Integrator" is the role that was previously titled "Glue" in (Sheard 1996) and the change as well as the rationale for the change is captured below. Tables 4-6 also highlight the roles framework developed, consisting of three categories:

- ***Roles Focused on the System Being Developed*** – These roles are what may most quickly come to mind when describing a systems engineer. They are roles that align closely with the systems engineering lifecycle and the critical activities systems engineers must enable throughout the lifecycle.
- ***Roles Focused on SE Process and Organization*** – These roles focus on the organizational context in which systems engineering occurs and the critical role of systems engineers in providing guidance on how systems engineering should be utilized.
- ***Roles Focused on Teams that Build Systems*** – Systems engineering does not occur in a vacuum and is, instead, an intensely social discipline. The roles in this category are those that focus on enabling diverse, multi-disciplinary teams to be successful.

The categories are intended to help draw distinctions between the major types of activities systems engineers provide.

Table 4. Roles Focused on the Systems Being Developed

Role Name	Role Description
Concept Creator	Individual who holistically explores the problem or opportunity space and develops the overarching vision for a system(s) that can address this space. A major gap pointed out to the Helix team – particularly when working to implement the findings of Helix – has been that of the development of an overarching system vision. This is a critical first step in the systems lifecycle, and several organizations stated that they believed it needed to be separated out. In addition, when looking to the future of what systems engineers need to do (e.g. INCOSE Vision 2025 (2015)), the focus on early engagement and setting the vision was deemed critical.
Requirements Owner	Individual who is responsible for translating customer requirements to system or sub-system requirements; or for developing the *functional* architecture. This is unchanged from (Sheard 1996).

Role Name	Role Description
System Architect	Individual who owns or is responsible for the architecture of the system. This is an update of Sheard's "System Designer" role (1996). There was concern both at community events and during later interviews that nowhere in the presented framework did the critical role of systems engineers in architecture come out clearly. Some also argued that "Design" gave the impression that this roles focuses specifically on the details of systems design over architecture.
System Integrator	Individual who provides a holistic perspective of the system; this may be the 'technical conscience' or 'seeker of issues that fall in the cracks' – particularly, someone who is concerned with interfaces. Likewise, there was concern over the word "Glue", which many expressed was not clearly descriptive enough.
System Analyst	Individual who provides modeling or analysis support to system development activities, and helps to ensure that the system as designed meets he specification. This is unchanged from Sheard's roles (1996).
Detailed Designer	Individual who provides technical designs that match the system architecture; an individual contributor in any engineering discipline who provides part of the design for the overall system. This is an addition based on the Helix data. While systems engineers do not always get involved with detailed design, in smaller organizations or on smaller projects it is more common. Likewise, systems engineers who had played this role explained that it was critical in developing their own technical and domain expertise as well as in understanding the design approaches of classic engineers.
V&V Engineer	Individual who plans, conducts, or oversees verification and validation activities such as testing, demonstration, and simulation. This is unchanged from Sheard's roles (1996).
Support Engineer	Individual who performs the 'back end' of the systems lifecycle, who may operate the system, provide support during operation, provide guidance on maintenance, or help with disposal. This was previously titled "Logistics and Operations Engineer" in Sheard (1996). However, in interviews and at community events, the Helix team received feedback that using this title gave the impression that this role was limited and did not encompass the full spectrum of systems engineers' activities at system deployment or post-deployment. Likewise, in several organizations, "logistics" and "operations" were seen as separate disciplines from systems engineering, which caused some contention in discussions. The renaming of this category is intended to address these issues.

Table 5. Roles Focused on Process and Organization

Role Name	Role Description
Systems Engineering Champion	Individual who promotes the value of systems engineering to individuals outside of the SE community - to project managers, other engineers, or management. This may happen at the strategic level or could involve looking for areas where systems activities can provide a direct or immediate benefit on existing projects. Sheard recommended that a role such as this, labeled in her work as "Systems Engineering Evangelist", be added in (2000).
Process Engineer	Individual who defines and maintains the systems engineering processes as a whole and who also likely has direct ties into the business. This individual provides critical guidance on how systems engineering should be conducted within an organization context. This is unchanged from Sheard's roles (1996).

Table 6. Roles That Focus on Teams That Build Systems

Role Name	Role Description
Customer Interface	Individual who coordinates with the customer, particularly for ensuring that the customer understands critical technical detail and that a customer's desires are, in turn, communicated to the technical team. This is unchanged from Sheard's roles (1996).
Technical Manager	Individual who controls cost, schedule, and resources for the *technical* aspects of a system; often someone who works in coordination with an overall project or program manager. This is unchanged from Sheard's roles (1996).
Information Manager	Individual who is responsible for the flow of information during system development activities. This includes the systems management activities of configuration management, data management, or metrics. This is unchanged from Sheard's roles (1996).
Coordinator	Individual who brings together and brings to agreement a broad set of individuals or groups who help to resolve systems related issues. This is a critical aspect of the management of teams. This is unchanged from Sheard's roles (1996).
Instructor/Teacher	Individual who is provides or oversees critical instruction on the systems engineering discipline, practices, processes, etc. This can include the development or delivery of training curriculum as well as academic instruction of formal university courses related to systems engineering. While any discipline could conceivably have an instructor role, this denotes a focus on systems and is a critical component in the development of an effective systems engineering workforce. This is an addition to the Sheard roles (1996)

The role of "Classified Ad" systems engineer was used by Sheard (1996) as a placeholder to acknowledge the many job postings for "systems engineers" which actually reflected IT network or computer specialists (e.g. network systems engineer, IT systems engineer, or Microsoft systems engineer). In the Helix sample, none of the systems engineers for whom roles data was collected played this role, either currently or in the past. In addition, when this role was presented at various community events (Helix workshops in 2014, 2015, and 2016; presentations on Helix at INCOSE (Lipizzi, 2015, Jauregui, 2016), there was a strong recommendation to remove it from the framework to highlight what systems engineers do and to draw a clear distinction from positions that may be titled "systems engineer" but which do not bear resemblance to the practice of systems engineering.

Data on roles can be found in Sections 4.5.1 and 5.1.3. Note that in the analyses for *Atlas 1.0*, the roles of *Concept Creator* and *Systems Engineering Champion* are not included. This is because, as new additions, the data for these roles was not collected consistently throughout the sample. Additional follow-up work will be required to ensure adequate and consistent data on these roles.

4.3.1 OTHER ROLES SYSTEMS ENGINEERS PLAY

Tables 4-6 outline the *systems engineering* roles. However, there were a few roles that were commonly seen throughout the Helix data sample. These are roles that may frequently be played by systems engineers. These include:

- *Organizational/Functional Manager* - Individual who is responsible for the personnel management of systems engineers or other technical personnel in a *business* – not a project or program – setting.
- *Program/Project Manager* - Individual who is not *directly* responsible for the technical content of a program, but works closely with technical experts and other systems engineers while

maintaining overall project cost and schedule.

These roles, while not systems engineering roles, are things that many systems engineers do throughout their careers and which may help systems engineers develop some critical skills. Figure 12 provides a simple Venn diagram showing, from the Helix data, the overlap between systems engineering roles and roles held by systems engineers.

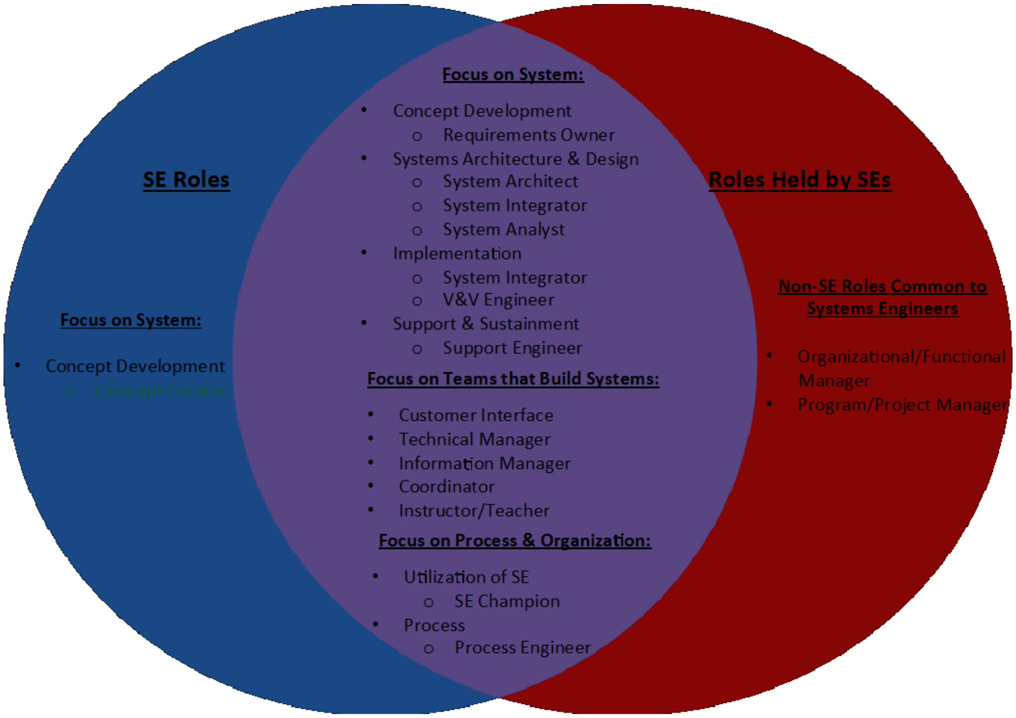


Figure 12.The overlap between SE roles and Roles Held by Systems Engineers in the Helix Sample

It may be surprising that one of the SE roles shown in Figure 12, "Concept Creator", is not a role that systems engineers in the Helix sample commonly played. A small number of individuals in the Helix sample did play these roles, but not enough, initially, to add these to the framework. The addition of these roles was based on community feedback and work on implementation with several organizations. The Helix team believes that the primary reason that "Concept Creator" did not come out of the sample is due to the organizations in which they work. In each of the government organizations that has participated, the systems engineers have been part of the acquisition workforce. When asked if they participated in initial concept definition, most explained that this was done before they were assigned to the system. Systems engineers at many industry organizations, particularly those within the DIB, expressed a similar view – that this early vision-setting happened before systems engineers got involved.

In looking to the future of systems engineers, there is a push for them to be included more in concept design. Clearly, concept development work is part of systems engineering as it is critical for successful systems, and one would assume that this would be an important role for systems engineers. This is reflected in strategic documents such as INCOSE *Vision 2025* (2015) as well as in the goals and desires of several organizations working to implement Helix findings and individual systems engineers. This is the rationale for inclusion in the *Atlas* roles framework.

4.3.2 IMPLICATIONS FOR USE

Individuals who are assessing their careers may benefit from the exercise of reviewing the roles which they have and have not played at various stages throughout their careers. (Templates in Appendix A.) This analysis will enable an individual to identify any clear patterns in their careers (e.g. a role that they have never played or a role that they have played consistently). This self-awareness about their own career will enable them to make more informed decisions about their future desired positions (e.g. identifying opportunities to play new roles).

There are several ways the roles assessments can benefit organizations. First, by enabling individuals to be more aware about their own careers, there may be some organic growth and development across the workforce. Second, understanding holistically any gaps or common patterns across the workforce may provide insight into potential Organizational Development Initiatives. For example, if only a small percentage of individuals have played the role of "Concept Creator", the organization can then determine (a) whether this is an appropriate role for their context and (b) if it is, can identify more clearly opportunities for systems engineers to play this role.

Organizations may also choose to use the role descriptions and profiles to increase clarity across the workforce in several ways. For example, if the roles profiles are created across the careers of individuals in key systems engineering positions (e.g. chief systems engineer, systems engineering lead, systems architect, etc.), then those patterns may provide some guidance for career planning at the individual and workforce levels. This is not to say that there will only be one "right way" to grow, but clear patterns from the organization paired with self assessment may help an individual identify which roles they would like to focus on going forward. A second way organizations may use the roles is in clarifying the open positions they have within the organization. In the Helix sample, it was quite rare for a position to consist of only a single role. (See analyses in Section 5.2) But several participating organizations stated that the current job positions – whether posted internally or externally – make it difficult for individuals to truly understand what is entailed. However, if the roles are clearly defined and used consistently, they could become a new model for describing what is expected in positions that would provide clarity for both individuals seeking the roles and individuals who are selecting for the roles.

4.4 PROFICIENCY OF SYSTEMS ENGINEERS

The proficiency model in *Atlas*, captures the knowledge, skills, abilities, behaviors, patterns of thinking, and abilities that are critical to the effectiveness of systems engineers. The key changes from the *Atlas 0.5* proficiency model, and the rationale for these changes, are described in detail below. However, the following is short list of the changes:

- ***System's Domain and Operational Context*** – the categories in this area have been reorganized to reflect more mature thinking as a result of the implementation work by the Helix team. A new category – *System Characteristics* – was also added. This contains characteristics that were previously under the Systems Engineering Discipline area.
- ***Systems Engineering Discipline*** – as noted above, the previous category *System Complexity* was moved to the System's Domain and Operational Context area, combined with other system attributes, and renamed. In addition, a new category, *Systems Engineering Trends*, has been added. This is a category in which to reflect key trends in the discipline, such as model-oriented systems engineering, agile systems engineering, etc., as appropriate.
- ***Technical Leadership*** – two new categories have been added to reflect a future vision for systems engineers. These categories, *Establishing Technical Strategies* and *Enabling Broad Portfolio-Level Outcomes*, reflect a vision of systems engineers not only for individual projects or programs but as individuals who will help set the technical vision for an organization.

4.4.1 GENERIC PROFICIENCY MODEL

The various elements of proficiency gathered are aggregated into a 3-level hierarchy, as illustrated in Figure 13 below:

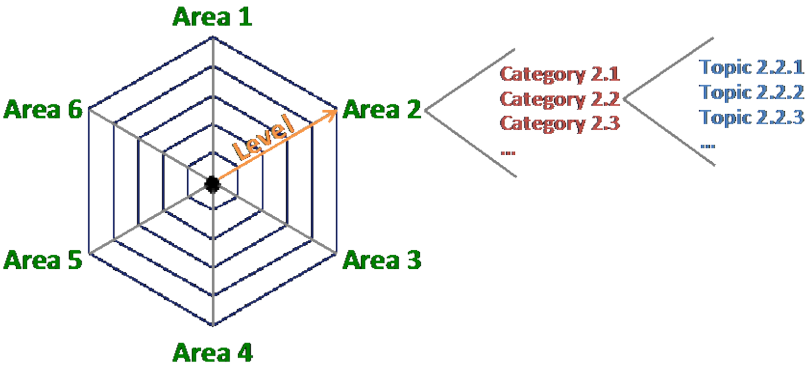


Figure 13. *Atlas* Proficiency Framework

- **Proficiency** is the quality or state of knowledge, skills, abilities, behaviors, and cognition.
- Proficiency **Areas** are groupings of related knowledge, skills, abilities, behaviors, and/or cognition.
- Each Proficiency Area is comprised of **Categories**, which are specific types of knowledge, skills, abilities, behaviors, and cognition with shared characteristics.
- *Some* categories are further refined into **Topics**, which are the most discrete areas of proficiency included in *Atlas*.

- For each proficiency area, there are **Levels**, which describe the extent to which an individual has attained certain knowledge, has the ability to perform a certain skill, or has demonstrated relevant abilities, behaviors, or cognition. Loosely, a scale of 1 to 10 is used to indicate the level of proficiency at the area level, where 10 indicates the highest possible proficiency. These scales will be further developed in *Atlas 1.0*.

The *Atlas* proficiency model along with the proficiency levels, enable a proficiency profile to be created for an individual at any point in time, as illustrated in Figure 14. Currently, proficiency levels are being considered only at the Area level.

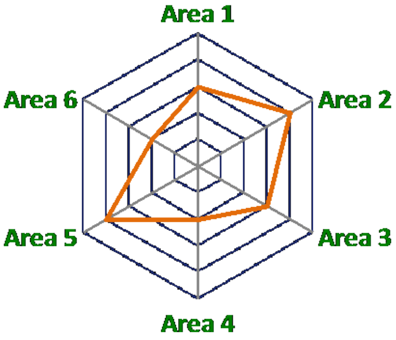


Figure 14. Sample Proficiency Profile

4.4.2 *ATLAS* PROFICIENCY MODEL

The *Atlas* proficiency model consists of six proficiency areas based on the Helix interview data, as shown in Figure 15 below.

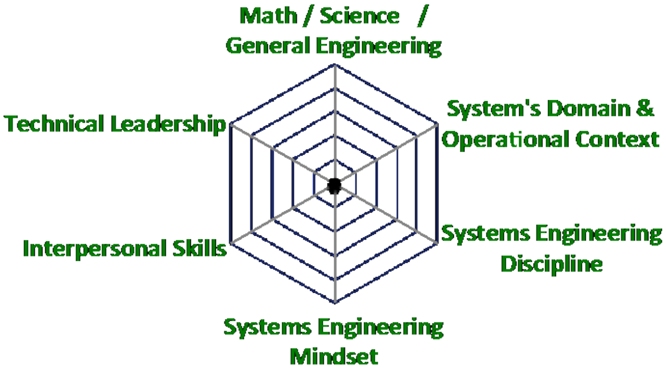


Figure 15. Proficiency Areas for Systems Engineers

1. ***Math/Science/General Engineering:*** Foundational concepts from mathematics, physical sciences, and general engineering;
2. ***System's Domain & Operational Context**:* Relevant domains, disciplines, and technologies for a given system and its operation;
3. ***Systems Engineering Discipline***: Foundation of systems science and systems engineering

knowledge;

4. ***Systems Engineering Mindset***: Skills, behaviors, and cognition associated with being a systems engineer;
5. ***Interpersonal Skills***: Skills and behaviors associated with the ability to work effectively in a team environment and to coordinate across the problem domain and solution domain; and
6. ***Technical Leadership***: Skills and behaviors associated with the ability to guide a diverse team of experts toward a specific technical goal.

Proficiency areas 1 to 3 consist of primarily 'hard' or technically based skills, while proficiency areas 4 to 6 consist primarily of the 'soft' or interdisciplinary skills. Development and evaluation of soft skills is addressed by the disciplines of psychology, social sciences, and management sciences. The six proficiency areas in *Atlas* are further divided into categories and, in some cases, into topics, as shown in Table 7. Each of the proficiency areas is elaborated in the subsequent sections.

Table 7. Atlas Proficiency Areas, Categories, and Topics

Area	Category	Topic
1. Math / Science / General Engineering	**1.1.** Natural Science Foundations	
	1.2. Engineering Fundamentals	
	1.3. Probability and Statistics	
	1.4. Calculus and Analytical Geometry	
	1.5. Computing Fundamentals	
2. Systems' Domain & Operational Context	**2.1.** Principal and Relevant Systems	< List of Principal and Relevant Systems >
	2.2. Familiarity with Principal System's Concept of Operations (ConOps)	
	2.3. Relevant Domains	< List of relevant Domains >
	2.4. Relevant Technologies	< List of relevant Technologies >
	2.5. Relevant Disciplines and Specialties	< List of relevant Disciplines and Specialties >
	2.6. System Characteristics	< List of applicable System Types, Scales, and Levels >
3. Systems Engineering Discipline	**3.1.** Lifecycle	3.1.1 Lifecycle Models 3.1.2 Concept Definition 3.1.3 System Definition 3.1.4 System Realization 3.1.5 System Deployment and Use 3.1.6 Product and Service Life Management
	3.2. Systems Engineering Management	3.2.1 Planning 3.2.2 Risk Management 3.2.3 Configuration Management 3.2.4 Assessment and Control 3.2.5 Quality Management
	3.3. SE Methods, Processes, and Tools	3.3.1 Balance and Optimization 3.3.2 Modeling and Simulation 3.3.3 Development Process 3.3.4 Systems Engineering Tools
	3.4. Systems Engineering Trends	3.4.1 Complexity 3.4.2 Model Oriented Systems Engineering 3.4.3 Systems Engineering Analytics 3.4.4 Agile Systems Engineering

Area	Category	Topic
4. Systems Engineering Mindset	**4.1.** Big-Picture Thinking	
	4.2. Paradoxical Mindset	**4.2.1** Big-Picture Thinking and Attention to Detail **4.2.2** Strategic and Tactical **4.2.3** Analytic and Synthetic **4.2.4** Courageous and Humble **4.2.5** Methodical and Creative
	4.3. Flexible Comfort Zone	
	4.4. Abstraction	
	4.5. Foresight and Vision	
5. Interpersonal Skills	**5.1.** Communication	**5.1.1** Audience **5.1.2** Content **5.1.3** Mode
	5.2. Listening and Comprehension	
	5.3. Working in a Team	
	5.4. Influence, Persuasion and Negotiation	
	5.5. Building a Social Network	
6. Technical Leadership	**6.1.** Building and Orchestrating a Diverse Team	
	6.2. Balanced Decision Making & Rational Risk Taking	
	6.3. Guiding Stakeholders with Diverse/Conflicting Needs	
	6.4. Conflict Resolution & Barrier Breaking	
	6.5. Business and Project Management Skills	
	6.6. Establishing Technical Strategies	
	6.7. Enabling Broad Portfolio-Level Outcomes	

4.4.3 AREA 1: MATH/SCIENCE/GENERAL ENGINEERING

A good understanding of math, science, and general engineering is a critical foundation for effective systems engineers; but this understanding is largely ‘assumed’ in a systems engineer when joining the workforce since proficiency in this area is not utilized directly or in isolation. However, it is upon this foundation that further understanding of the categories under Proficiency Area 2: *Systems’ Domain & Operational Context* is built.

The *Graduate Reference Curriculum for Systems Engineering (GRCSE®)* defines the types of prerequisite knowledge individuals should have before entering a master’s program in systems engineering (Pyster et al. 2015). Since limited insight was obtained from Helix data collection and analysis for this proficiency area, GRCSE is used to identify and define the categories in this area:

1.1. **Natural Science Foundations:** Basic concepts and principles of one of the natural science disciplines (e.g., physics, biology, chemistry, etc.); includes laboratory work that involves experimental techniques, the application of the scientific method, and comprehension of appropriate methods for data quality assurance and analysis.

1.2. **Engineering Fundamentals:** The nature of engineering, branches of engineering, the design process, analysis and modeling, the role of empirical and statistical techniques, problem solving strategies, and the value of standards; some level of practical experience is expected, whether through capstones, internships, or course projects. Practical experience should include the application of engineering fundamentals in a specific domain context.

1.3. **Probability and Statistics:** Basic probability theory, random variables and probability distributions, estimation theory, hypothesis testing, regression analysis, and analysis of variance.

1.4. **Calculus and Analytical Geometry:** Theory and application of differential and integral calculus methods and operations; study of techniques for describing, representing, and analyzing geometric objects (coordinate systems, algebraic models, graphing).

1.5. **Computing Fundamentals**: Overview of computer organization (computer architecture, operating systems, and programming languages), algorithms, and data structures; software engineering fundamentals (lifecycle models, quality, cost, and schedule issues); and development of a software unit (design, coding, and testing).

Proficiencies in Area 1: Math/Science/General Engineering may be considered as the general foundation that is provided in any undergraduate engineering degree. Advanced levels of these topics are included in the topics of Area 2, in the context of the system of concern. For an individual without a formal undergraduate degree in engineering, obtaining the proficiencies in Area 1 could happen through experience, training, or mentoring.

4.4.4 AREA 2: SYSTEM'S DOMAIN & OPERATIONAL CONTEXT

The second proficiency area is *System's Domain & Operational Context*, which contains the relevant domains, technologies, disciplines, specialties, and characteristics for a given system, and the operation of that system. This proficiency area strongly corresponds to the organization and the systems that its systems engineers work on. If an individual transitions to a new system either, the proficiency level may change depending on familiarity with the new relevant domains, technologies, and disciplines. The categories for this proficiency area are defined below:

2.1. **Principal and Relevant Systems**: *Principal* systems are those systems that are of primary interest to the organization. High levels of proficiency in those specific systems are desired by the organization. If a combat ship were the principal system, relevant systems could be submarines and aircraft carriers, which are types of combat ships.

2.2. **Familiarity with Principal System's Concept of Operations (ConOps)**: A system's concept of operations (ConOps) of how systems in the domain are used and deliver value, especially those systems on which the individual personally works. Familiarity with the principal system's ConOps is of particular interest, though familiarity with the ConOps of other related systems may also be helpful.

2.3. **Relevant Domains**: *Domain* refers to the overarching area of application of the system; this includes things such as space, aerospace, marine, communication, finance, etc. Proficiency in related domains outside the primary one may enable an individual to be more effective in the primary domain. For example, experience in space systems may enable a systems engineer to work in aerospace systems more readily than an engineer who is proficient primarily in finance systems.

2.4. **Relevant Technologies**: Within the context of a system, there are specific technologies that are relevant. For example, on a marine system, these may be technologies such as gas turbine, radar, and sonar systems; and each technology has its own terminology, challenges, etc.

2.5. Relevant Disciplines and Specialties: Disciplines are fundamental areas of education or expertise that are foundational to a system. For example, for a communications system, electrical engineering will be an important discipline to understand, while civil engineering will be less relevant. Specialties are disciplines that support systems engineering by applying cross-cutting knowledge. Specialties include Reliability, Availability, and Maintainability (RAM), Human Systems Integration, Safety Engineering, Affordability and other related topics.

2.6. System Characteristics: Three characteristics are considered in *Atlas*:

- **System Type:** Types of systems include technical systems, social systems, human systems, physical systems, cyber systems, and any combination of these. Another classification of system types includes product systems, service systems, and enterprise systems.
- **System Scale:** Systems can be anywhere from a nano level to a distributed global or enterprise level. A generic systems engineering development process may be applicable to systems at any scale.
- **System Scope:** What can be seen as a system from one perspective, could be a subsystem from another perspective. The levels of a system could range from component/element, subsystem, system, and platform or system of systems.

Category 2.6 is a change from older versions of *Atlas*. In previous versions, there was a category called "System Complexity" under Proficiency Area 3 (Systems Engineering Discipline). However, as the Helix team worked with organizations that were implementing, questions arose about other systems aspects in the proficiency model. For example, there was a concern that system complexity was identified but not the scope of a system nor different system scopes from element to system of systems. The Helix team agreed that though these were covered in the experiences of systems engineers, this was a gap in the existing proficiency model. After reviewing discussions on proficiency from the data, the team determined that several characteristics should be called out. Because these were related to the types of systems, the team determined that the new category, "System Characteristics", was a better fit in Proficiency Area 2 (System's Domain and Operational Context).

Note that this refers to the ability of an individual to work effectively on systems with different characteristics. This is related to, but distinct from, experiences an individual has on these types of systems. The experiences are discussed in Section 4.5, below.

4.4.5 AREA 3: SYSTEMS ENGINEERING DISCIPLINE

The third proficiency area is *Systems Engineering Discipline*. The categories below were developed based on data from Helix interviews about critical systems engineering knowledge and skills. The names of the categories come from the *Guide to the Systems Engineering Body of Knowledge (SEBoK)* (BKCASE Editorial Board 2016). Some of the categories are further expanded into topics.

3.1. Lifecycle: *The organized collection of activities, relationships and contracts that apply to a system-of-interest during its life* (Pyster 2009). This is a roll up of knowledge about lifecycles and proficiency in specific aspects of the lifecycle. Topics 3.1.2 – 3.1.6 below, represent generic lifecycle phases in system development:

3.1.1. Lifecycle Models: *A framework of processes and activities concerned with the lifecycle that may be organized into stages, which also acts as a common reference for communication* *and understanding* (ISO/IEC/IEEE 15288). Lifecycle Models include Vee model; iterative models such as the spiral development model; formal acquisition models (e.g., as defined in DoD 5000.2 2013); or less formal acquisition models (e.g., quick reaction capability or internal research and development (IR&D) models).

3.1.2. Concept Definition: *A set of core technical activities of systems engineering in which the problem space and the needs of the stakeholders are closely examined* (BKCASE Editorial Board 2016). This consists of analysis of the problem space, business or mission analysis, and the definition of stakeholder needs for required services.

3.1.3. System Definition: *A set of core technical activities of systems engineering, including the activities that are completed primarily in the front-end portion of the system design.* (BKCASE Editorial Board 2016) This consists of the definition of system requirements, the design of one or more logical and physical architectures, and analysis and selection between possible solution options.

3.1.4. System Realization: *The activities required to build a system, integrate disparate system elements, and ensure that a system both meets the needs of stakeholders and aligns with the requirements identified in the system definition stage* (BKCASE Editorial Board 2016). This includes implementation as well as integration, verification, and validation (IV&V).

3.1.5. System Deployment and Use: *A set of core technical activities of systems engineering to ensure that the developed system is operationally acceptable and that the responsibility for the effective, efficient, and safe operations of the system is transferred to the owner* (BKCASE Editorial Board 2016). Considerations for deployment and use must be included throughout the system lifecycle. Activities within this phase include deployment, operation, maintenance, and logistics.

3.1.6. Product and Service Life Management: Deals with the overall lifecycle planning and support of a system (BKCASE Editorial Board 2016). The life of a product or service often spans a considerably longer period of time than the time required to design and develop the system. This stage includes service life extension, updates, upgrades, and modernization, and disposal and retirement.

3.2. Systems Engineering Management: *Managing the resources and assets allocated to perform systems engineering, often in the context of a project or a service, but sometimes in the context of a less well-defined activity. Systems engineering management is distinguished from general project management by its focus on the technical or engineering aspects of a project* (BKCASE Editorial Board 2016). The topics contained in the *Systems Engineering Management* category are defined below:

3.2.1. Planning: Planning involves developing and integrating technical plans to achieve the technical project objectives within the resource constraints and risk thresholds. This involves the success-critical stakeholders to ensure that necessary tasks are defined with the right timing in the lifecycle in order to manage acceptable risks levels, meet schedules, and avoid costly omissions (BKCASE Editorial Board 2016).

3.2.2. Risk Management: Organized, analytic process to identify what might cause harm or loss (identify risks); to assess and quantify the identified risks; and to develop and, if needed, implement an appropriate approach to prevent or handle causes of risk that could result in significant harm or loss (ISO/IEC/IEEE 24765:2010 – SEVocab).

3.2.3. Configuration Management: A discipline applying technical and administrative direction and surveillance to: identify and document the functional and physical characteristics of a configuration item, control changes to those characteristics, record and report change processing and implementation status, and verify compliance with specified requirements (ISO/IEC/IEEE 24765:2010 – SEVocab).

3.2.4. Assessment and Control: This process involves determining and initiating the appropriate handling strategies and actions for findings and/or discrepancies that are uncovered in the enterprise, infrastructure, or lifecycle activities associated with the project (BKCASE Editorial Board 2016).

3.2.5. Quality Management: Whether a systems engineer delivers a product, a service, or an enterprise, the deliverable should meet the needs of the customer and be fit for use. Such a deliverable is said to be of high quality. The process to assure high quality is called quality management (BKCASE Editorial Board 2016).

3.3. SE Methods, Processes, and Tools: *A systems engineering method is set of activities, methods, practices, and transformations that people use to develop and maintain systems and associated products* (SEI 2007). Processes generally refer to the specific guidelines an organization develops for implementing systems engineering methods; tools refer to software programs that are designed to support systems engineering activities. The topics contained in the *SE Methods, Processes, and Tools* category are outlined below:

3.3.1. Balance and Optimization: Specialty engineers often focus on the details and optimization of their specific components of the system, but that optimization of individual components often leads to a less-than-optimal system solution. Systems engineers, therefore, have to be able to balance the desire for component optimization with the optimization for the system overall, which often requires sub-optimization for one or more components.

3.3.2. Modeling and Simulation: *A model is a simplified representation of a system at some particular point in time or space intended to promote understanding of the real system. A simulation is the manipulation of a model in such a way that it operates on time or space to compress it, thus enabling one to perceive the interactions that would not otherwise be apparent because of their separation in time or space* (Bellinger 2004). This topic represents and individual’s ability to understand and perform modeling and simulation; this understanding is more fundamental than the ability to use software tools that support modeling and simulation.

3.3.3. Development Processes: Each organization has its own processes that govern the development of systems. It is important for systems engineers to understand generic systems engineering processes, but also the specific processes being used for development within the organization or domain.

3.3.4. Systems Engineering Tools: Systems engineers need to be able to utilize tools to support overall system development and to perform the systems engineering development process. Tools may include requirements management and other tools that assist with project life management (PLM).

3.4. Systems Engineering Trends: *Current and future trends in performing Systems Engineering, that modify the way systems are developed.*

3.4.1. Complexity: Complexity of a system is generally understood to exist not in a higher order scale or level of a system, but rather in the higher order of interactions between system elements, disciplines, or technologies, and the properties that emerge out of these interactions that are not present in the individual elements. One categorization of complexity includes structural complexity, dynamic complexity, and socio-political complexity; while another identifies two kinds of complexity: disorganized complexity and organized complexity (SEBoK authors 2016).

3.4.2. Model Oriented Systems Engineering: Model Based Systems Engineering (MBSE) is a theme that is being increasingly adopted in systems engineering, where models are used to describe various elements of systems and the systems development process. Model Oriented Systems Engineering (MOSE) goes beyond MBSE, and presents a holistic model-based approach that integrates operational, technical, programmatic and business dimensions as well.

3.4.3. Systems Engineering Analytics: The increasing ability to collect, store, analyze, and gain insights from large quantities of data has significantly improved the area of analytics in general. This perspective can also be applied to systems engineering, where complex phenomena within systems and systems development can be measured and analyzed.

3.4.4. Agile Systems Engineering: The shrinking of systems engineering development lifecycles, increasingly uncertain and rapidly changing requirements and operational environments of modern systems, has led to the development and adoption of agile systems engineering approaches.

Category 3.4 is a change from older versions of *Atlas*. As discussed in Section 4.4.5, above, the in previous version this category simply covered system complexity. However, as the Helix team worked with organizations trying to implement *Atlas*, there were several questions about "trendy" topics in systems engineering and where they should fit within the proficiency model. The Helix team determined that as these are specific applications of systems engineering, they fit within Proficiency Area 3; the Helix team has labeled these "trends". Rather than selecting a single trend, the Helix team determined that creating a category for "Systems Engineering Trends" was more reasonable than creating categories for each potential trend. The trends listed in 3.4 are consistent with some of the areas of interest expressed across a number of organizations in the Helix sample. Additional trends could be added as they become more prominent in the systems engineering community.

4.4.6 AREA 4: SYSTEMS ENGINEERING MINDSET

The fourth proficiency area is *Systems Engineering Mindset*, which is primarily focused on patterns of thinking, perceiving, and approaching a task that are particularly relevant to systems engineers. The categories included in this area are:

4.1. Big-Picture Thinking: Also referred to as 'systems thinking' and 'holistic thinking', this includes the ability to step back and take a broader view of the problem at hand; this is an important and essential characteristic of systems engineers. 'Big-picture' could refer to a broader perspective along many different dimensions: the system as a whole including interfaces and integration, and not limited to any sub-system or component; the system while in operation, and its interactions with other systems and the operating environment; the entire lifecycle of the system, and not limited to the current stage of the system; the development program in the context of the organization and all its other development programs; the end goal or solution to the problem at hand; the perspectives of different stakeholders; and the technical as well as business perspectives. A systems engineer is usually *the* person to bring this broader perspective, while classic engineers and subject matter experts often tend to be narrowly focused on their area of interest. Systems engineers are not only called to provide this big-picture perspective themselves, but to also enable others to see this bigger picture.

4.2. **Paradoxical Mindset**: *The ability to hold and balance seemingly opposed views, and being able to move from one perspective to another appropriately*. Typically, an engineer may hold one view or the other, but rarely *both*. By having this paradoxical mindset, a systems engineer contributes value that is not usually expected from others. The opposing-concept pairs are:

   4.2.1. **Big-Picture Thinking *and* Attention to Detail:** Big-picture thinking provides the broader higher-level perspective; at the same time, a systems engineer is also required to pay attention to the details of how things work and how they come together in a system.

   4.2.2. **Strategic *and* Tactical:** Systems engineers need to be strategic, focused on the end result of 'vision' for the system, but also need to handle the tactical day-to-day activities and decisions required to reach that vision. They must also be able to appreciate "how what is done today is going to affect things downstream". A related concept pair is the ability to envision long-term issues but at the same time, have the desire for closure with the current situation in order to move on.

   4.2.3. **Analytic *and* Synthetic:** A big-picture perspective may be associated with the ability to be synthetic, and to be able to bring together and integrate different pieces of a puzzle. However, a systems engineer also needs to be analytic and to be able to break down the big picture into smaller pieces on which others can focus and work. To do this effectively, a systems engineer needs to be able to operate at multiple levels (e.g., component, sub-system, system, system-of-systems) and multiple dimensions (e.g., various technical disciplines and stakeholder perspectives).

4.3. **Flexible Comfort Zone**: *The overall ability to deal with ambiguity and uncertainty, this involves the abilities to be open-minded, understand multiple disciplines, deal with challenges, and the ability to take rational risks*. By definition, experts possess proficiency in a specific area, which is their 'comfort zone'; and they typically do not prefer going outside that circle or comfort zone. Such experts provide value to the organization by contributing their expertise in those focused areas. However, systems engineers tend to show an ability to broaden their comfort zones, and go beyond their current boundaries and they are also comfortable doing this.

4.4. **Abstraction**: *The ability to filter out and understand the critical bits of information at the right level and to make relevant inferences*. And even with that filtered information, systems engineers need to know when to use or not use pieces of information. Such abstraction also enables systems engineers to connect and extract meaning from different streams of information; for example, to tie together information that subject matter experts of two different disciplines are providing.

4.5. **Foresight and Vision**: *The ability to foresee the remaining lifecycle of the system, the impact of current decisions, and to mentally simulate possible scenarios*. Every decision or change is likely to have an impact beyond the current confines of time or space. Particularly in early stages of a system lifecycle, and in the development of a new or unfamiliar system, foresight is a key value that systems engineers provide.

4.4.7 AREA 5: INTERPERSONAL SKILLS

The fifth proficiency area is *Interpersonal Skills*. Almost by definition, systems engineers do not just work by themselves at their desks all day – they interact with people. Irrespective of any formal leadership roles they may or may not play, a systems engineer is expected to be proficient in a number of interpersonal skills. While specialty engineers may be responsible for developing specific aspects of the system, systems engineers are responsible for coordinating across all of these engineers. Hence, interpersonal skills are more critical to systems engineers than they are to specialty engineers. The specific categories contained within this proficiency area are listed below:

5.1. Communication: Communication is critical for systems engineers since they interact with a variety of people, and is a broad category covering a wide variety of related skills and abilities. Often they are an important link between individuals and groups, both internal and external to the organization – most importantly, the customers and end-users of the system being developed. Systems engineers need the ability to clearly express their thoughts and perspectives to establish a shared common understanding.

5.1.1. Audience: Systems engineers need to communicate with a variety of direct and indirect audiences: customers; subject matter experts; program managers; vice presidents; directors; specialty engineers; problem owners; technical teams; contractors; decision makers; system testers; and others working on or with the project.

5.1.2. Content: The variety of content that systems engineers need to communicate can be broadly divided into three types, based on the audience they are communicating with:

1. **Technical**: Communications with disciplinary and specialty engineers and subject matter experts involve high technical content. But communications of technical issues to managers, end-users, and others who may not be interested in or who may be confused by all the technical detail, involves adequate abstraction of the technical content.

2. **Managerial**: Systems engineers often provide project status to managers and supervisors and cost-schedule constraints and expectations to technical personnel.

3. **Social**: Systems engineers need to maintain an amicable environment within a team and to interact with others in a courteous manner. Such interactions involve communications that are neither technical nor managerial in nature.

5.1.3. Mode: Communicating the intended content to the target audience is done through a number of different modes:

1. **Oral:** This takes various forms, depending on the audience and context. It could be one-on-one, or as part of a team, in person, or remotely.

2. **Presentation**: A special form of communications is the ability to stand in front of an audience and to deliver a presentation using appropriate aids. Further, during presentations, systems engineers tend to represent others who may not be in the room: they present customer needs and requirements to others in the absence of customers, and they present design decisions and system related issues to customers in the absence of designers.

3. **Writing and Documentation:** Written communication skills are equally critical for systems engineers; the scale, audience, and objective of the written artifact also matter. It could range from a short email to communicate status, to a detailed test plan, to internal documentation supporting a project decision, to design documents being submitted for review.

5.2. Listening and Comprehension: The ability to listen to others' points of views and perspectives, and to comprehend and internalize the message accurately. For systems engineers, listening begins with the customer to understand their real needs and ensure that these needs get translated into requirements. In a team environment, systems engineers need to listen to the views and perspectives being offered: from designers, subject matter experts, and others.

5.3. Working in a Team: Systems engineers tend to be part of many teams during the lifecycle of the system; further, systems engineering by itself is typically not performed by an individual, but rather by a team. Hence, team dynamics and synergy are key to the functioning of a systems engineer.

5.4. Influence, Persuasion, and Negotiation: It is critical for every systems engineer, not just those in formal leadership positions, to have the skills needed to make a point and to successfully obtain buy-in. In many situations, systems engineers contribute a perspective that is different from that of others: a focus on the overall system and on customer's needs. In such situations, it requires influence, persuasion, and negotiating skills for systems engineers to enable others to see the bigger picture on which they need to focus.

5.5. Building a Social Network: A systems engineer needs to be a 'people person', and build a social network of professional acquaintances. Such a network becomes a valuable resource for systems engineers to tap into, because they are not expected to know answers to all problems, but rather be able to find someone who has the expertise and ability to solve the problem.

4.4.8 AREA 6: TECHNICAL LEADERSHIP

The sixth and final *Atlas* proficiency area is *Technical Leadership*. It is common and natural for systems engineers to play leadership roles at many levels within an organization. The specific categories contained within *Technical Leadership* are listed below:

6.1. Building and Orchestrating a Diverse Team: *The ability to identify, build, and effectively guide or coach a team comprising individuals with diverse expertise, perspectives, and personalities*. While organizational titles may vary, it is most often a systems engineer who is the leader of the team that is charged with delivering the system. The systems engineer needs to fully know each of the team members: their strengths, weaknesses, capacities, capabilities, limitations, personalities, expertise, and working styles. The systems engineer plays the roles of coach, guide, and teacher to develop the team's capabilities and to orchestrate it to perform the required tasks. Individual leadership styles could vary, but the overall objective of is to empower the team, to instill confidence, and to help them to deliver the solution and to be successful. Another key aspect of handling a team is the ability to delegate – the leader needs to build enough trust in the team to be able to delegate with confidence.

6.2. Balanced Decision Making and Rational Risk Taking: Solving a problem requires a systems engineer to take a number of balanced decisions considering a variety of factors, constraints, perspectives, and objectives; as well as the implications of these decisions and their scope of impact. An additional challenge is that most often, all the required information may not be readily available. The ability to make such decisions also requires the systems engineer to be comfortable in dealing with ambiguity and uncertainty and to be able to take rational, calculated risks.

6.3. Guiding Stakeholders with Diverse/Conflicting Needs: This includes the ability to manage all the internal and external stakeholders, and to keep the team focused on their needs, especially those of the end user or customer. The systems engineer is uniquely positioned to interact with many stakeholders of the system – both external and internal to the organization. Being this "touch point" person, the systems engineer needs to deal with multiple personalities, behaviors, organizations, and cultures.

6.4. **Conflict Resolution and Barrier Breaking:** Conflicts are bound to rise in a variety of scenarios – within the team; within the organization – between the technical side and business side of the organization; as well as with outside the organization. As a leader, the systems engineer must resolve these conflicts while keeping the system goals in mind. In some cases, conflicts arise due to the existence of barriers, which may be related to the organizational culture, processes, team personalities, or other situations that could prevent an individual or team from getting their work done. The systems engineer needs the ability to break these barriers.

6.5. **Business and Project Management:** Depending on the way roles and titles are defined within an organization, a systems engineer's responsibilities may overlap with what may be seen as 'project management' responsibilities. Even if there is no overlap, a systems engineer is expected to handle a variety of business and project management activities including accounting, budget, cost estimation, schedule, work breakdown, and profit. The systems engineer must also be cognizant of the business impact of technical decisions that are taken.

6.6. **Establishing Technical Strategies:** Systems engineers must fearlessly and creatively guide the establishment of new capabilities and transformations (e.g., to migrate to Cloud Infrastructure, or to establish a new information service architecture, or to enable transition to a DEVOPS model). Senior systems engineers need to be able to support the organization in the development of overarching technical directions and support the development of technical roadmaps that establish a vision to support the strategy.

6.7. **Enabling Broad Portfolio-Level Outcomes:** Along with the development of strategies to guide strategic technical investments, systems engineers should provide the broad perspective necessary to enable technical success not only on individual projects but across projects and programs to enable advancement across the technical portfolio.

Note that Categories 6.6 and 6.7 are new to *Atlas 1.0*. These categories speak not just to technical leadership within teams, but also to technical leadership within an organization. These were not initially included because they were heard clearly at few organizations in the Helix sample. However, the Helix team hopes that *Atlas* will be relevant not only today but in the future. To this end, community outreach, implementation work, and literature review in 2016 has focused on ensuring that the proficiency model will be relevant for future systems engineers as well. Categories 6.6 and 6.7 speak to a vision of roles that systems engineers should play in future (e.g. INCOSE *Vision 2025,* 2014) and aligns with proficiencies already expected of senior systems engineers in some organizations.

4.4.9 IMPLICATIONS FOR USE

Tailoring

As demonstrated in Table 7, there is a clear expectation that some tailoring will occur for proficiency assessment to maximize its utility. This is true for both individuals and organizations. Individuals may tailor the model specifically to what they have done – but should be mindful that all of the areas they have not touched are possible areas for future exploration. Organizations, likewise, could tailor the model before distributing it to the workforce, so that only areas that are deemed critical to the organization are captured. For example, some of the natural science foundations may not be common in a given domain and some disciplines or technologies will be considered more relevant than others. It is important to remember that tailoring may not be specific to just an organization, but also to specific programs or systems. For example, an organization that engineers financial IT systems as well as critical infrastructure systems may have different expectations and needs for those different domains.

Table 8 provides two examples of how the proficiency model could be tailored for an organization, based on the primary systems domain for each organization. Note that where <no tailoring> is listed, this indicates that the Helix team does not expect that an organization will be able to use the proficiency model exactly as defined, with no tailoring required.

Table 8. Tailoring the Atlas Proficiency Model

Area	Category	Company 1: Defense Aerospace	Company 2: Medical Devices
1. Math / Science / General Engineering	**1.1.** Natural Science Foundations	Physics considered most critical	Chemistry and Biology considered most critical Physiology added as a Foundation
	1.2. Engineering Fundamentals	<no tailoring>	<no tailoring>
	1.3. Probability and Statistics	<no tailoring>	<no tailoring>
	1.4. Calculus and Analytical Geometry	Both are considered critical	Considered less critical than Probability & Statistics
	1.5. Computing Fundamentals	Considered less critical than the other categories	Considered critical for integration with Electronic Health Records (EHRs)
2. Systems' Domain & Operational Context	**2.1.** Principal and Relevant Systems	Air-breathing jet engines Military aircraft	Magnetic Resonance Imaging (MRI) X-Ray Computerized Tomography (CT)
	2.2. Familiarity with Principal System's Concept of Operations (ConOps)	Expectations about the level of familiarity may differ (e.g. understanding basic in-flight operations)	Expectations about the level of familiarity may differ (e.g. actual experience in a clinical setting to understand use cases, how system fits within the healthcare environment, where its use may fit in an overall process, etc.)
	2.3. Relevant Domains	Aerospace	Healthcare
	2.4. Relevant Technologies	Radar Sonar Navigation Systems	MRI X-Ray CT
	2.5. Relevant Disciplines and Specialties	Mechanical Engineering Electrical Engineering Aerospace Engineering Software Engineering Thermodynamics Aerodynamics Ergonomics	Electrical Engineering Mechanical Engineering Biomedical Engineering Software Engineering Ergonomics Radiation Safety
	2.6. System Characteristics	System level design with understanding of the system of systems in the operational environment	Systems of systems level design enabling integration with other medical devices and healthcare IT systems
3. Systems Engineering Discipline	**3.1.** Lifecycle	• V- lifecycle approach emphasized • Organization not involved in in-service operation and maintenance (full	• Spiral/Incremental Development lifecycle model emphasized • Organization heavily involved in in-service operation and

Area	Category	Company 1: Defense Aerospace	Company 2: Medical Devices
		handoff after delivery)	maintenance
	3.2. Systems Engineering Management	<no tailoring>	<no tailoring>
	3.3. SE Methods, Processes, and Tools	• Heavy emphasis on Modeling and Simulation • Emphasis on operational safety	• Heavy emphasis in optimization for patient safety
	3.4. Systems Engineering Trends	• Model Oriented Systems Engineering	<no tailoring>
4. Systems Engineering Mindset	**4.1.** Big-Picture Thinking	<no tailoring>	<no tailoring>
	4.2. Paradoxical Mindset	• Balance of Methodical and Creative heavily weighted	• Paradoxical mindset heavily weighted
	4.3. Flexible Comfort Zone	<no tailoring>	<no tailoring>
	4.4. Abstraction	<no tailoring>	<no tailoring>
	4.5. Foresight and Vision	<no tailoring>	<no tailoring>
5. Interpersonal Skills	**5.1.** Communication	<no tailoring>	<no tailoring>
	5.2. Listening and Comprehension	<no tailoring>	<no tailoring>
	5.3. Working in a Team	<no tailoring>	<no tailoring>
	5.4. Influence, Persuasion and Negotiation	<no tailoring>	<no tailoring>
	5.5. Building a Social Network	<no tailoring>	<no tailoring>
6. Technical Leadership	**6.1.** Building and Orchestrating a Diverse Team	<no tailoring>	<no tailoring>
	6.2. Balanced Decision Making & Rational Risk Taking	<no tailoring>	Risk is viewed negatively by this highly safety-conscious organization; this becomes focused on decision making.
	6.3. Guiding Stakeholders with Diverse/Conflicting Needs	<no tailoring>	<no tailoring>
	6.4. Conflict Resolution & Barrier Breaking	<no tailoring>	<no tailoring>
	6.5. Business and Project Management Skills	• Project management is treated as a distinctly separate discipline from systems engineering in this organization. There is cultural pressure not to include this as a “systems engineering” proficiency.	<no tailoring>
	6.6. Establishing Technical Strategies	• N/A (Systems engineers do not set the technical strategy for the organization)	• Only expected for senior systems engineers
	6.7. Enabling Broad Portfolio-Level Outcomes	• N/A (Systems engineers do not set	• Only expected for senior systems

Area	Category	Company 1: Defense Aerospace	Company 2: Medical Devices
		the technical strategy for the organization)	engineers

Table 8 is only a basic example, but demonstrates that tailoring can include the identification of specific proficiencies that are of critical interest to the organization – particularly in Proficiency Areas 1 and 2, which are expected to be heavily tailored – and the emphasis or de-emphasis of categories based on the organizational context. The examples for categories 6.6 and 6.7 also demonstrate that the organization can help to set expectations about categories that are critical only at certain seniority levels.

Implications for Use by Individuals

The Helix team has assisted over 100 individuals in completing self-assessments based on these proficiencies and dozens of others have completed self-assessments in their organizations without the team's involvement.

During some of the Helix interviews in 2015, interviewees were asked to self-evaluate their level of proficiency based on the *Atlas* proficiency model, at the Area level. Generally, interviewees evaluated themselves on a level of 1 to 10, where 1 was 'least proficient' and 10 was 'most proficient'. This was a subjective scale and hence when someone placed themselves at an 8 for a proficiency area, for example, it was based on their personal interpretation on what it meant. These self-evaluations – and subsequent discussions on why interviewees scored themselves in a particular way – are expected to provide insights in future research towards defining those objective scales.

Interviewees were asked to evaluate their proficiencies at two points in time: (1) at the time of the interview, and (2) at the start of their career. This enables a proficiency profile to be plotted, as illustrated in Figure 16.

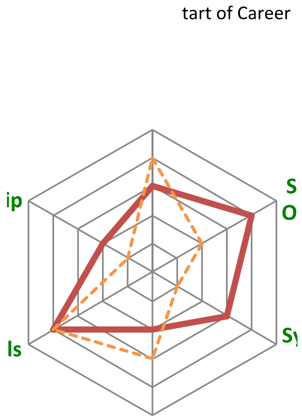


Figure 16. Proficiency Profile of an Individual

The proficiency profile is not meant to be exact since the self-evaluations are subjective, and individuals may have over-evaluated or under-evaluated themselves. Also, 'Start of Career' could be as recent as five years ago for one individual or twenty-five years ago for another. However, this exercise enables a discussion around the relative strengths in specific proficiencies; how proficiency levels changed over time; and what factors or forces caused or enabled those changes.

The primary intent of *Atlas* is not to just understand the current state of effective systems engineers, but to support the development of future systems engineers who will be effective. From a proficiency perspective, it would mean setting target levels for proficiency areas, as illustrated in Figure 17.

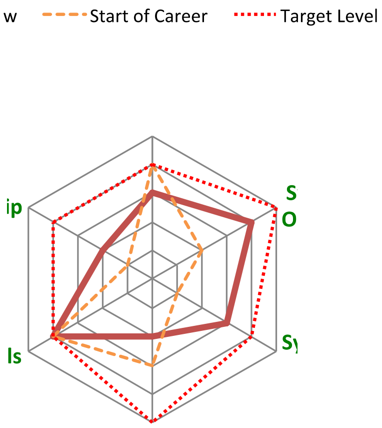
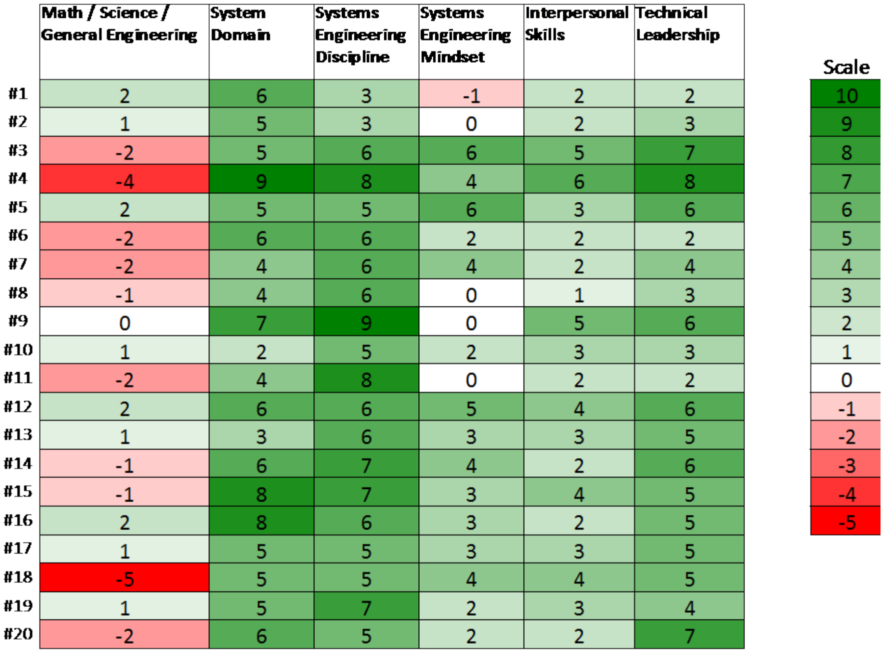


Figure 17. Proficiency Profile with Target Levels

Identifying target levels for the proficiencies will depend on the roles or positions that that individual aspires to play in future. For a junior or mid-level systems engineer, the target level could be based on the proficiency profile of a Chief Systems Engineer. This profile of a CSE is also influenced by the expectations of the organization. Having proficiency profiles, including target levels, similar to what is shown in Figure 17, would enable individuals to identify those proficiencies that need to be developed and by how much. Individual systems engineers could then plan their career development in a more focused and intentional manner, towards a specific goal.



	Math / Science / General Engineering	System Domain	Systems Engineering Discipline	Systems Engineering Mindset	Interpersonal Skills	Technical Leadership
#1	2	6	3	-1	2	2
#2	1	5	3	0	2	3
#3	-2	5	6	6	5	7
#4	-4	9	8	4	6	8
#5	2	5	5	6	3	6
#6	-2	6	6	2	2	2
#7	-2	4	6	4	2	4
#8	-1	4	6	0	1	3
#9	0	7	9	0	5	6
#10	1	2	5	2	3	3
#11	-2	4	8	0	2	2
#12	2	6	6	5	4	6
#13	1	3	6	3	3	5
#14	-1	6	7	4	2	6
#15	-1	8	7	3	4	5
#16	2	8	6	3	2	5
#17	1	5	5	3	3	5
#18	-5	5	5	4	4	5
#19	1	5	7	2	3	4
#20	-2	6	5	2	2	7

Scale
10
9
8
7
6
5
4
3
2
1
0
-1
-2
-3
-4
-5

Figure 18. Example Workforce Profile: Change in Proficiency Levels of Individuals

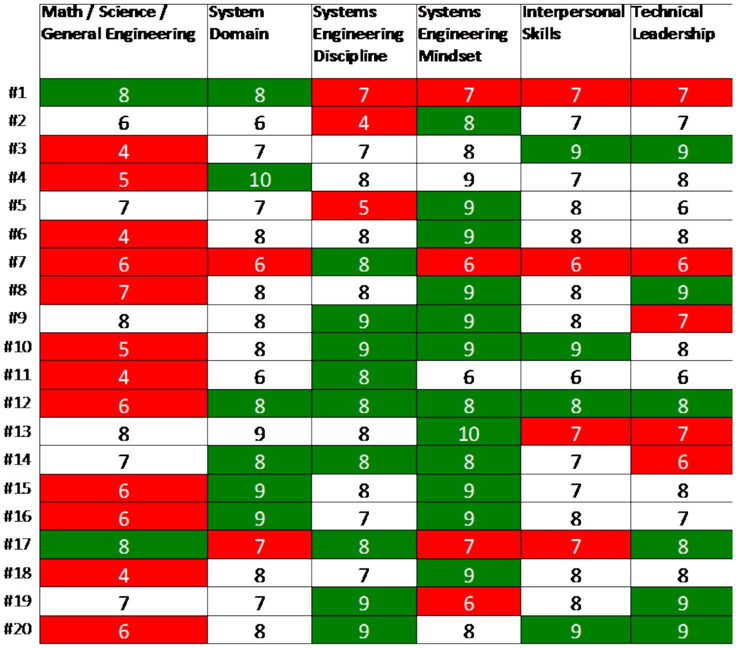
Figure 18 illustrates a typical distribution of changes in proficiency levels in systems engineers who participated in Helix interviews. Again, the intent is not to perform a detailed quantitative statistical analysis, but to use the information to gather insights for career development:

- The *Math/Science/General Engineering* area is one where many systems engineers said their proficiency levels dropped during their careers. One of the main reasons stated was that they were not using those skills nearly as often (or at all) in their current roles as they did at the start of their careers.
- Systems engineer #4 saw big improvements in the *System Domain*, *SE Discipline*, and *Technical Leadership* areas. Insights into the factors that contributed to those improvements will benefit others who wish to improve those proficiencies.
- Systems engineers #9 and #11 saw big improvements in the *SE Discipline* area, but did not have any change in their level of proficiency in the *SE Mindset* area. Exploring the reasons for this could reveal fresh insights.
- Overall, improvements in the *Interpersonal Skills* area are observed to be relatively modest. It will be useful to explore the reasons behind this.

Implications for Use by Organizations

Developing the career of an individual systems engineer necessitates the concerned individual to make decisions and take required actions. However, they can be done only in the context of the organization. For example, an individual may identify the need for a master's degree in systems engineering as critical to developing some much-needed proficiencies. But if the organization does not encourage or enable its employees to pursue higher education, that systems engineer may not be able to obtain a master's degree while employed in that organization. Hence, there is a critical role that an organization plays in developing the careers of individual systems engineers.

There are insights that an organization may be able to obtain by studying the collective proficiency profiles of its systems engineers. Figure 19 shows the self-assessment of the same 20 random systems engineers included in Figure 18, for all six of the *Atlas* proficiency areas, highlighting the strongest (green) and weakest (red) proficiency areas.



	Math / Science / General Engineering	System Domain	Systems Engineering Discipline	Systems Engineering Mindset	Interpersonal Skills	Technical Leadership
#1	8	8	7	7	7	7
#2	6	6	4	8	7	7
#3	4	7	7	8	9	9
#4	5	10	8	9	7	8
#5	7	7	5	9	8	6
#6	4	8	8	9	8	8
#7	6	6	8	6	6	6
#8	7	8	8	9	8	9
#9	8	8	9	9	8	7
#10	5	8	9	9	9	8
#11	4	6	8	6	6	6
#12	6	8	8	8	8	8
#13	8	9	8	10	7	7
#14	7	8	8	8	7	6
#15	6	9	8	9	7	8
#16	6	9	7	9	8	7
#17	8	7	8	7	7	8
#18	4	8	7	9	8	8
#19	7	7	9	6	8	9
#20	6	8	9	8	9	9

Figure 19. Example Workforce Profile: Strongest and Weakest Proficiencies of Individuals

If the systems engineering population represented in Figure 19 were to belong to a single systems engineering team or group, studying it could help the team or group recognize workforce development issues as well as opportunities as identified below:

- *Math/Science/General Engineering* is the weakest area for most systems engineers. Of greater interest than identifying reasons for this trend is exploring the impacts of this on the organization’s systems engineering capability.
- The *SE Mindset* and *SE Discipline* areas are the strongest for many systems engineers, but also the weakest for some. There may be an opportunity here to establish some mentoring initiatives focusing on these proficiency areas.

The *Technical Leadership* area is a mixed bag, with almost equal number of systems engineers saying it is their strongest or weakest proficiency area. Based on further exploration, a training course could be established within the organization focusing on the specific aspects of technical leadership where systems engineers feel they are the weakest.

4.4.10 INTERPRETATION OF PROFICIENCY DATA

It is important to note that to date, though there is a rubric to guide proficiency self assessment, the numbers for a proficiency level are more comparative than strictly quantitative; i.e., they are useful to understand how an individual views their changes in proficiency over time. In a group setting, participants were also able to discuss what they meant by their proficiencies and by comparing these with those of their peers, make adjustments so that within that group, the numbers became more consistent.

However, at this point it would be unwise to over-interpret the numbers used for proficiency assessment, particularly when comparing proficiency levels across a number of individuals. The greater value to date is the conversation around the self-assessment. One organization that has been working to implement *Atlas* uses the proficiency in exactly this way – as a prompt for discussion between individuals and their leaders to inform career planning.

In 2017, the Helix team plans to use implementation opportunities as a way to further refine the rubrics for proficiency levels so that they become more consistent and comparable.

4.5 Forces that impact the Proficiency of Systems Engineers

The three most important forces that significantly impact the proficiency of systems engineers are *Experiences*, *Mentoring*, and *Education & Training*, in that order. These forces are generated by a combination of personal and organizational initiatives. The application of these forces is the primary way by which proficiencies of an individual are developed, as illustrated in Figure 20 below.

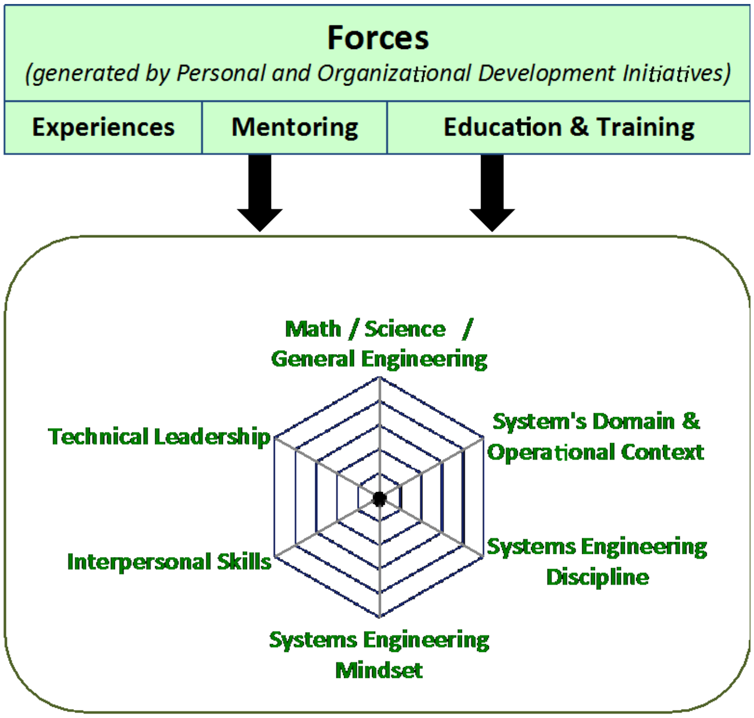


Figure 20. Forces and Proficiency

Insights into these forces that were identified from Helix data, and their relevance and importance for systems engineers, are discussed in Sections 4.5.1 through 0.

4.5.1 FORCE 1: EXPERIENCES

Experiences are considered the most critical factor contributing to the development of proficiencies and to the overall growth of systems engineers. However, it is the characterization of these experiences that provides insight into how they impact proficiencies over time. Considering experiences as a force, each of these dimensions contributes to increasing one or more areas of proficiency. Experiences can also impact the personal characteristics of an individual. *Experiences*, as considered in *Atlas*, includes experiences along the following characteristics:

- **Relevance:** Every experience cannot be considered to be relevant to the development of systems engineers. A 'relevant' position is one that enables a systems engineer to develop the proficiencies critical to systems engineering. A 'systems engineering' position is one where the individual's primary focus was on SE activities.
- **Position:** Every systems engineer who is employed at an organization fills a position that is established by the organization; that organization also defines the roles and responsibilities to be performed. Helix considers position as a 'unit of measure' for experience, since most of the characteristics of experience is in the context of the position that is being held.
- **Chronological Time:** The amount of time spent in any particular position or in performing a role.
- **Number of Organizations:** The number of different organizations that an individual has worked at, not counting internal movement within an organization across departments or divisions, reflects the variety of experiences that one may possess. In large corporations that have multiple business units, or in situations where there are mergers and acquisitions, this number may not be a good indicator of the variety of experiences.
- **Organizational Sectors:** There are many differences in the general characteristics of an organization based on its sector. In *Atlas*, three organizational sectors are identified: government, industry, and academia.
- **Roles:** The 16 roles identified in *Atlas* are described in Section 4.2.
- **Lifecycle Phases:** Generic systems engineering lifecycle phases considered in *Atlas* are described in Section 4.4.5. The titles and descriptions of lifecycle phases or stages may vary across different systems engineering processes and frameworks available in literature or in use at an organization.
- **Systems:** There are many aspects to the types of systems on which a systems engineer could work. Working across these different categories provides valuable experience to an individual systems engineer.
    - **Domain:** This is the primary area of application for the systems being worked on. However, there are many domain categorizations; some domains also relate to industry sectors.
    - **Type:** Product systems, service systems, and enterprise systems are three major types of systems, depending on the nature and composition of the system of interest. System of systems is another paradigm in systems engineering, and could be a combination of one or more types of systems.
    - **Scope:** A systems engineer could work on various levels of a system: component/element, subsystem, system, and platform or system of systems.

Analysis of Experiences

When asked for their recommendations on how systems engineers *should* grow and mature, 62% of the feedback captured in excerpts focused on the need for breadth of experiences and indicated this was critically important for growth. While 24% of the excerpts focused on the required depth of experiences in a particular discipline, technology, or domain, 23% of the excerpts on depth also described the need to balance "enough" depth with breadth of experiences.

Relevant Experiences

In Helix, the experiences of an individual that are considered for analysis are only the ones that are 'relevant' to systems engineering – that is, those that enable the proficiencies critical to systems engineering to be developed. It was observed in the careers of interviewees that the first few positions in their careers may have developed their professional skills, but did not specifically develop their systems engineering skills.

A 'systems engineering' position is a 'relevant' position where the primary focus is on systems engineering activities, irrespective of the title. Many interviewees identified their first systems engineering position in hindsight, since they were not aware at that time that they were performing systems engineering activities, or the organization did not officially recognize systems engineering activities.

Understanding the timing of the first relevant position and first systems engineering position enables a common 'starting point' to be defined for systems engineers, which then makes it easier to compare or develop career paths. However, it was not always easy to identify the duration of relevant experiences in an interviewee's career since they were discussing initial stages of their career that could be decades long. Figure 21 below illustrates the years of relevant experience of systems engineers in the Helix sample.

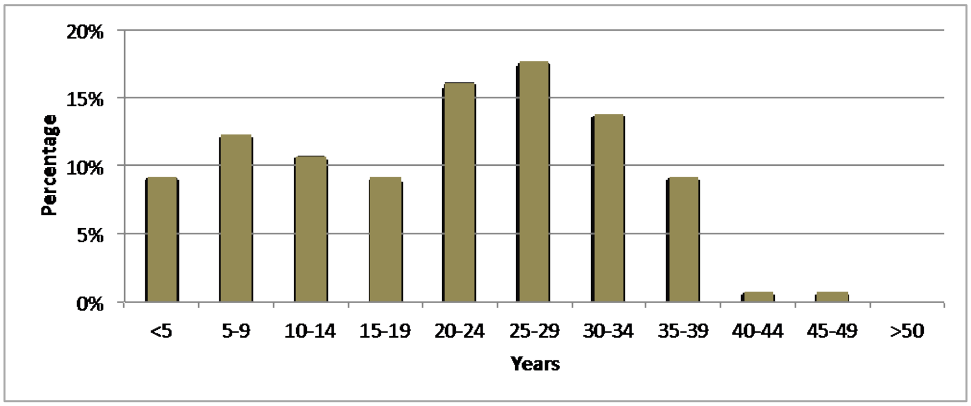


Figure 21. Years of Relevant Experience for The Systems Engineers in The Helix Sample

The distribution in Figure 21 shows that over two-thirds of the sample (68%) have 20 years or more of relevant experience, while 17% of systems engineers have less than 10 years of relevant experience. Years of experience cannot be directly correlated to proficiency, but clearly a longer range of relevant experiences may provide a systems engineer with more opportunities to build proficiencies. The years of relevant experience segregated by seniority is illustrated below in Figure 22. It must be noted that years of relevant experience does *not* necessarily correspond to the age of the systems engineer or the duration of their entire careers.

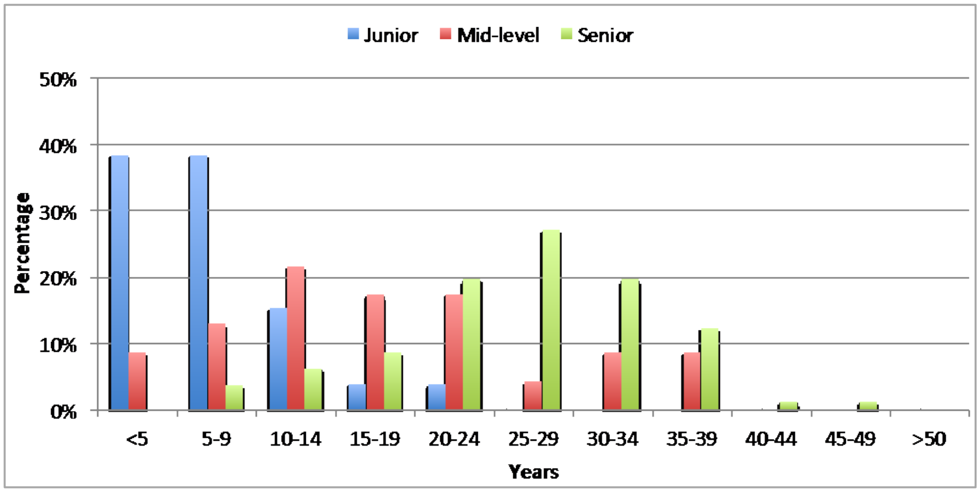


Figure 22. Years of Relevant Experience, by Seniority

Figure 22 illustrates several findings regarding the years of experience across the Helix sample:

- In the range of 5 to 24 years of relevant experience, there are interviewees who were at all levels of seniority.
- Over 75% of junior systems engineers have fewer that 10 years of relevant experience.
- Among mid-level systems engineers, about 70% have between 5 and 20 years of experience.
- There are mid-level systems engineers with over 30 years of experience, but they have not yet taken on leadership roles in systems engineering that would make them senior systems engineers.
- Senior systems engineers have between 8 and 47 years of experience, but with nearly two-thirds (64%) having between 20 and 40 years of experience.
- Senior systems engineers with over 40 years of experience are individuals who are eligible for retirement, but who stated in their interviews that they still enjoy their work and see no reason to leave.
- Senior systems engineers with 10 or less years of experience tend to be individuals who were spoken of by managers and other interviewees as “high potential” or “individuals to watch”.

The most significant observation from Figure 22 is that ‘years of relevant experiences’ is not a reliable metric to identify the seniority of a systems engineer; and for this reason, has not been included as a criterion for distinguishing the seniority of systems engineer in Table 2 in Section 7.

The total number of years of relevant experience or the number of years spent in any particular position or role has a direct impact on the depth or level of relevant proficiencies. However, even though more time would improve familiarity, it is also possible that the impact diminishes or stagnates over time. Hence, considering just chronological time is not valuable in understanding the impact on proficiencies.

Experiences Across Organizations

When an individual works within a single organization for a long period of time, she learns and internalizes the organization's processes for systems engineering, builds a network of peers that they leverage to better perform systems engineering, and how to operate within the organization. All of these things contribute to a systems engineer's proficiency and effectiveness. However, moving to a new organization provides opportunities for gaining new proficiencies. Exposure to different processes or systems engineering approaches helps systems engineers better understand the conditions appropriate to different approaches, and improves their ability to tailor processes and approaches as appropriate. Working within a new culture provides opportunities to better understand the impacts of culture on the overall effectiveness of systems engineers. Though transitions might be difficult, they can provide valuable experiences.

Figure 23 shows the distribution of total number of organizations worked across the sample, divided by seniority.

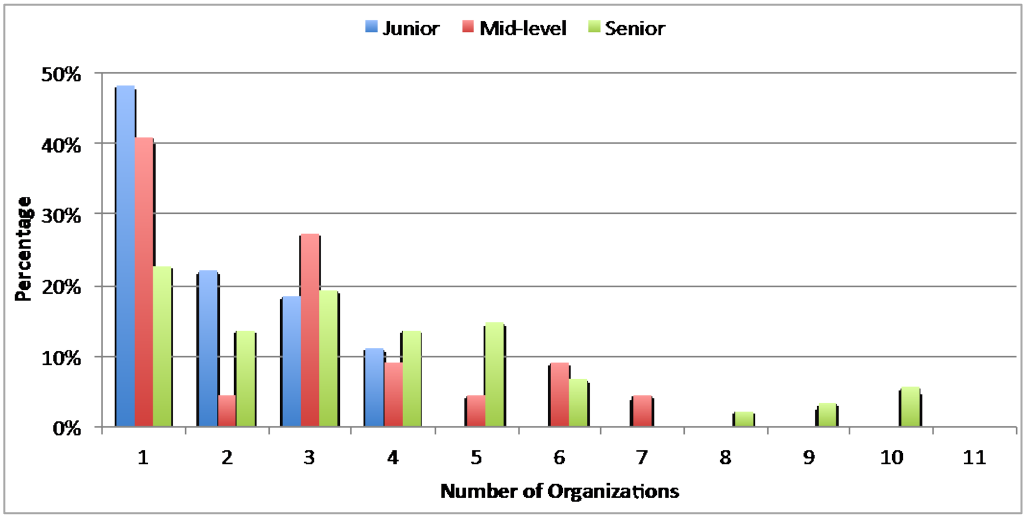


Figure 23. Experiences across Organizations, by Seniority.

The following observations can be made from Figure 23:

- Only the most senior systems engineers in the sample have worked in 8 or more organizations, as they generally have had the longest careers and, therefore, the most opportunities for movement between organizations.
- Nearly 50% of the junior systems engineers have worked in only one organization, as their careers have been generally much shorter.
- Over 40% of mid-level systems engineers and over 20% of senior systems engineers have only worked within a single organization. Those who fall within this category explained that they understood the organizational context so well and are satisfied with that context, that they see no need to make changes.

Number of Organizational Sectors

In terms of individual experiences, it is useful to understand whether an individual has worked for government, industry, or academic organizations, or a combination of these. At the time of their interview, 29% of the sample was currently working in government and 71% was currently working in industry. No interviewees in the Helix sample were currently working in academia alone.

In general, government systems engineers tend to oversee work done by systems engineers in industry as opposed to having the same direct responsibility for a system as seen in industry. There was one government organization in the current sample in which systems engineers reported directly performing hands-on systems engineering development work rather than primarily overseeing the work of others.

The different types of positions in the different sectors provide opportunities to develop new proficiencies – perhaps in new domains or operational contexts, or perhaps in new ways that systems engineering would be applied. However, some individuals stated that it might be difficult to transition between sectors because the overall ways in which the organizations operate, the processes used, and the cultures embedded, may be nearly polar opposites. This may mean that some skills become either obsolete or even harmful in a new organizational sector.

Moving across organizational sectors provides the opportunity to build new proficiencies. While this may be true for movement between any organizations, interviewees indicated that the impact is significant when moving between organizations in different sectors.

Figure 24 shows the distribution of the organizational sectors that Helix interviewees have worked in so far during their systems engineering relevant careers.

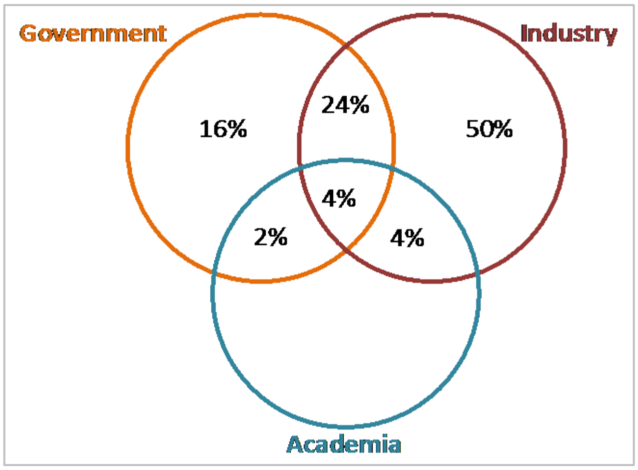


Figure 24. Variety of Organizational Sectors Experienced by Systems Engineers

The following observations can be made from Figure 24:

- 50% of the systems engineers in the sample have worked only in industry during their careers, while 16% have worked only in government.
- Almost a quarter have moved between industry and government during their careers, and in both directions; there is no pattern observed in the direction of movement. Interestingly, individuals who have moved in either direction often cite the same reasons for the movement: stability and a perceived opportunity for increase in technical responsibilities.
- Only 4% of the interviewees have industrial and academic experience, while only 2% have government and academic experience.

- There were a few individuals (4%) who had experiences in all three sectors.

<u>Experiences Across Roles Played</u>

Atlas identifies 16 roles for systems engineers, as listed in **Error! Reference source not found.**. The roles that an individual actually gets to play are dependent on the positions she occupies as defined by the organization. It is common for multiple roles to be performed within a single position. Single-role positions are typically encountered early or late in one's career. When found early in the career, systems engineers focused on detailed work in one role, but when found late in the career, systems engineers performed in roles such as management or teaching. In order to be effective in those late-career roles, earlier experiences in a variety of roles are useful.

Figure 25 shows the distribution of total number of roles played by systems engineer across the sample, divided by seniority.

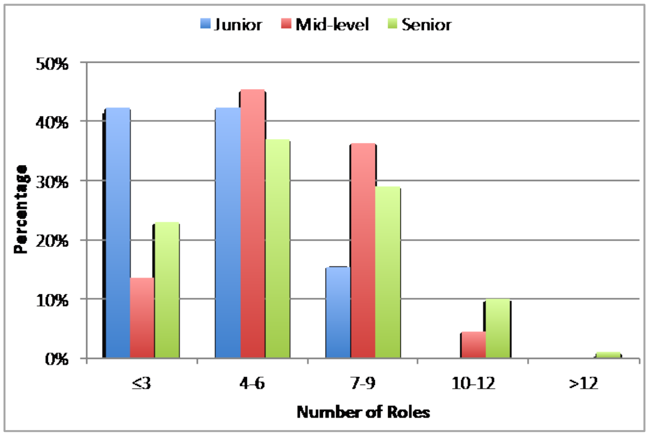


Figure 25. Total Number of Roles Played by Systems Engineers, by Seniority.

The following observations can be made from Figure 25:

- Over 40% of junior systems engineers have played up to 3 roles.
- Over 50% of junior systems engineers have played 4 to 9 roles. This is largely due to junior systems engineers participating in rotational assignments offered by organizations to new hires or newly-selected systems engineers.
- Over 75% of mid-level systems engineers have played 4 to 9 roles. Again, many of these individuals participated in rotational programs as junior systems engineers, increasing the number of roles played.
- Some senior systems engineers have played more than 12 roles across their entire career; almost all of these were also chief systems engineers.

Figure 25 also shows that there is no consistent grouping that can be used to differentiate the seniority levels of systems engineers. For this reason, 'number of roles played' is not included in the criteria listed in Table 2.

Figure 26 shows the percentage of systems engineers of the sample who have performed the various roles identified in *Atlas*. Note that the roles of *Concept Creator* and *Systems Engineering Champion* are not included in the figure. This is because, as new roles, the data for these roles was not collected consistently throughout the sample. Additional follow-up work will be required to adequately and consistently collect data on these roles.

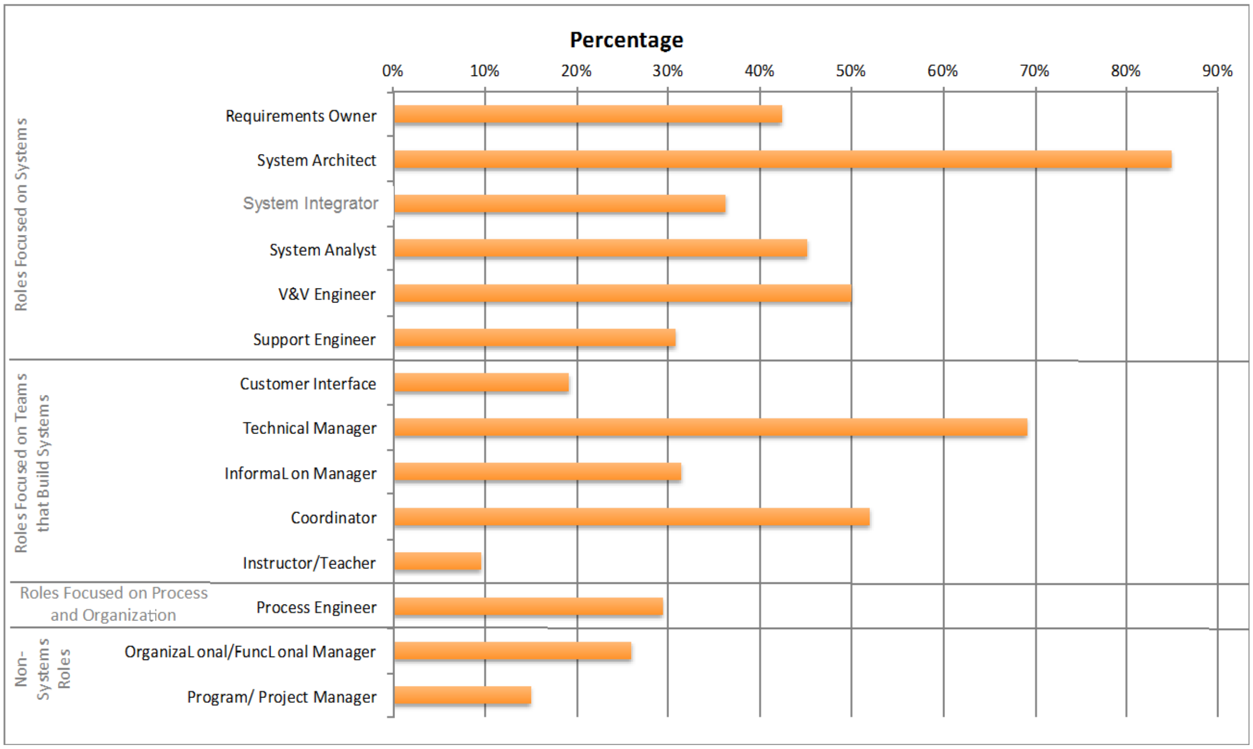


Figure 26. Distribution of Roles Played by Systems Engineers in the Helix Sample

Based on Figure 26 and other analyses on roles from interview data and resumes of interviewees, the following observations can be made:

- It was more common that multiple roles were played within a single position than only one type of role was played within a single position. For example, the 23 chief systems engineers (CSEs) in the Helix sample held a total of 279 positions and only 12% of the positions consisted of a single role. Of the single-role positions, 44% occurred early in a career (Position 4 or earlier). The remanding were primarily design work or instructor positions.
- The roles *Requirements Owner*, *System Designer*, *System Analyst*, *V&V Engineer*, and *Operations and Logistics Engineer* tend to occur earlier in a systems engineers' career, particularly when they occur as single roles in a position.
- The roles *Technical Manager*, *System Integrator*, *Coordinator*, *Customer Interface*, and *Process Engineer* tend to occur later in a systems engineer's career.
- It was common for systems engineers to take on additional roles later in their careers that may not traditionally be considered "systems engineering" roles. These roles include *Organizational/Functional Manager* – often within a systems engineering organization; *Program/Project Manager* – either concurrent with or separate from systems engineering responsibilities; and *Instructor/Teacher* – specifically of systems-related training or education.

Experiences across Lifecycle Phases

A variety of experiences across systems lifecycle phases provides critical experience to an individual, who gets to learn and 'feel' the impacts of decisions – the intended and unintended consequences of early-lifecycle decisions on the system during later lifecycle phases. *Atlas* uses the generic systems engineering lifecycle phases described in Section 4.4.5.

Because organizations use different terminology and models for lifecycles – e.g. 15288 (2015), DoD 5000.2 (2013), FAA Systems Engineering Process (2015) – the Helix team collected raw data on lifecycle phases and normalized each of them to the lifecycle phases as described in the *Guide to the Systems Engineering Body of Knowledge* (2016):

- **Concept Definition** - A set of core technical activities of SE in which the problem space and the needs of the stakeholders are closely examined. This consists of analysis of the problem space, business or mission analysis, and the definition of stakeholder needs for required services within it.
- **System Definition** - A set of core technical activities of SE, including the activities that are completed primarily in the front-end portion of the system design. This consists of the definition of system requirements, the design of one or more logical and physical architectures, and analysis and selection between possible solution options.
- **System Realization** - The activities required to build a system, integrate disparate system elements, and ensure that a system both meets the needs of stakeholders and aligns with the requirements identified in the system definition stage. This includes integration, verification, and validation (IV&V).
- **System Deployment and Use** - A set of core technical activities of SE to ensure that the developed system is operationally acceptable and that the responsibility for the effective, efficient, and safe operations of the system is transferred to the owner. Considerations for deployment and use must be included throughout the system life cycle. Activities within this stage include deployment, operation, maintenance, and logistics.
- **Product and Service Life Management** - Deals with the overall life cycle planning and support of a system. The life of a product or service spans a considerably longer period of time than the time required to design and develop the system. This stage includes service life extension, updates, upgrades, and modernization, and disposal and retirement. The organizations in the current sample are primarily concentrated on new development, so this is a very under-represented aspect of the life cycle.
- In addition to these life cycle phases, the SEBoK includes orthogonal activities of systems engineers, **Systems Engineering Management**, defined as managing the resources and assets allocated to perform SE activities. Activities include planning, assessment and control, risk management, measurement, decision management, configuration management, information management, and quality management. These activities can occur at any point in the systems engineering lifecycle.

Figure 27 shows the distribution of total number of lifecycle phases experiences by systems engineers across the sample, divided by seniority.

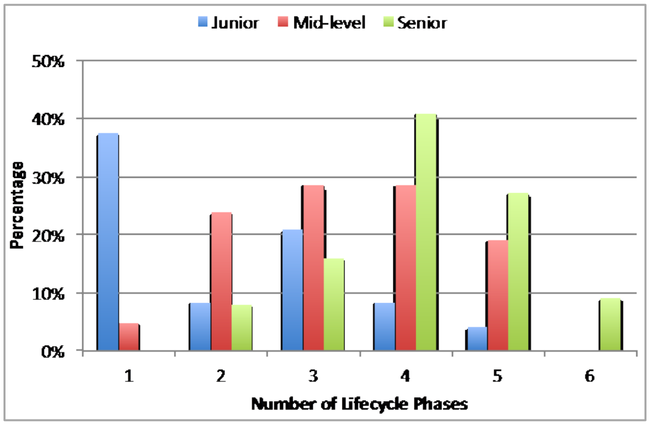


Figure 27. Total Number of Lifecycle Phases Experienced by Systems Engineers, by Seniority

Figure 28 shows the order in which systems engineers in the Helix sample were exposed to the various lifecycle phases, along their career.

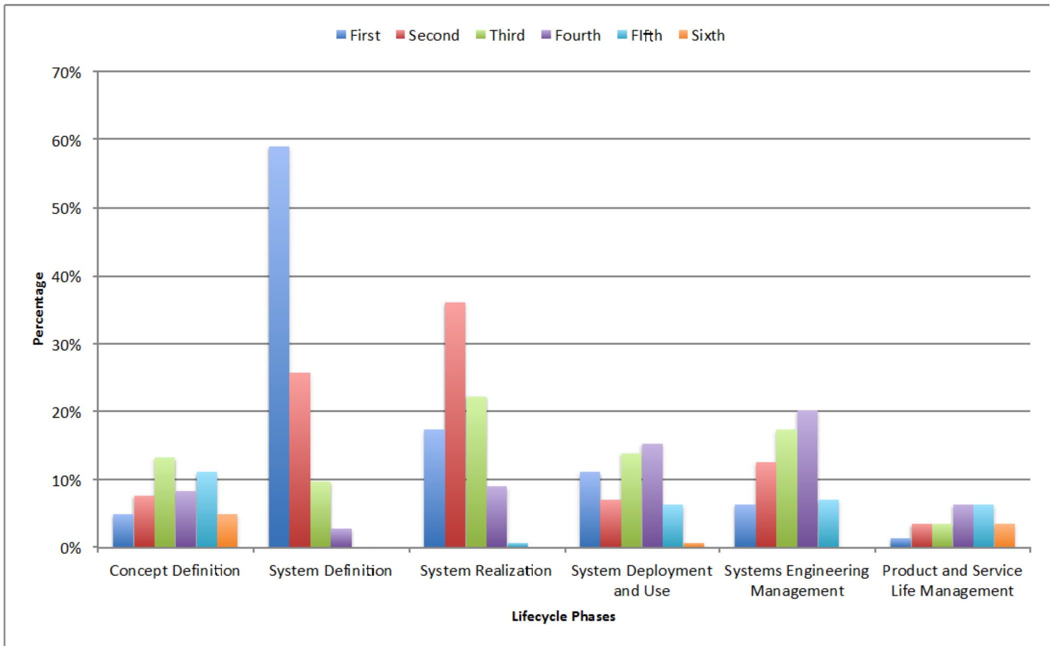


Figure 28. Order of Exposure to Lifecycle Phases, Experienced by Systems Engineers

The following observations can be made from Figure 27 and Figure 28:

- Only about 8% of the systems engineers in the Helix sample have had exposure to all lifecycle phases, all of whom were senior systems engineers.
- Over 50% had experiences in 4 or 5 phases of the lifecycle. The majority were senior or mid-level systems engineers. The few junior systems engineers who had seen this level of variety in the lifecycle had participated in rotational programs.
- The least commonly seen activity is *Product and Service Life Management*, which deals primarily with late lifecycle system activities. This may be a reflection of the organizations currently in the

Helix sample, that tend to be more focused on early lifecycle activities.

- Within the sample, there were 86 distinct paths through the systems lifecycle. Nearly 60% of the sample began their careers in System Definition. System Realization was the second lifecycle phase experienced by roughly a third of the sample.

When asked if there was an "ideal" path through the lifecycle, individuals were heavily influenced by the path that *they* had taken through the lifecycle. Individuals who had their start in system definition consistently said that this was the "best" place to start systems learning, giving experience with requirements and architecture. Those who started in System Realization described out verification and validation activities helped them understand the mistakes that could happen in design and how considerations for V&V activities should be incorporated into earlier lifecycle activities. Overall, there was not data in the Helix sample that indicated that there was an "ideal" way to move through the systems engineering lifecycle.

System Domain, Type, and Level

System Domain

The Helix team did not encounter a standard set of domains for systems engineering application in a review of the literature. The words used by interviewees to describe the domains in which they had worked were captured. Based on this, the Helix team used the North American Industry Classification System (NAICS) to create reasonable groups of domains. Experiences across these domains, divided by seniority, is illustrated in Figure 29. This distribution is only a reflection of the systems engineers currently in the Helix sample.

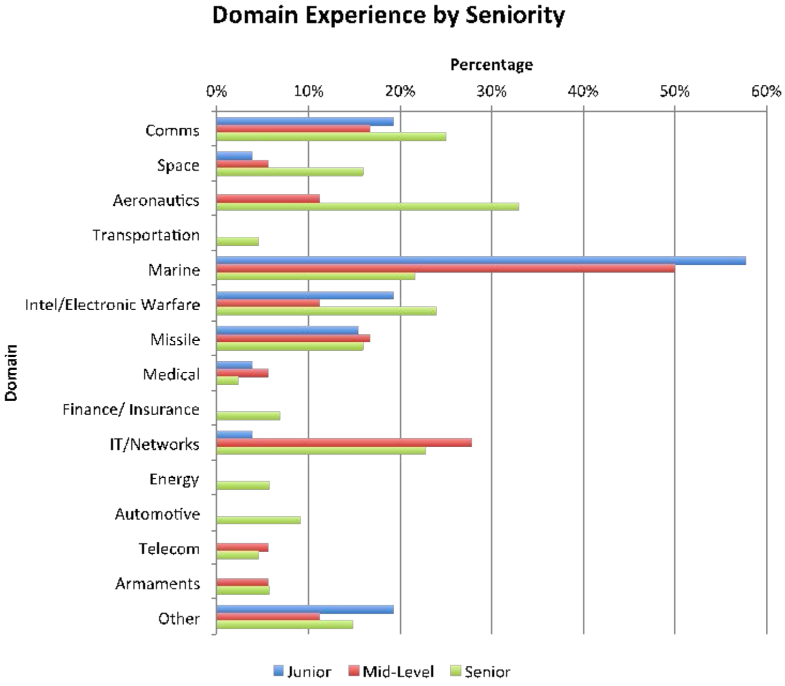


Figure 29. Experiences of Systems Engineers across System Domains, by Seniority.

System Type

All systems engineers in the Helix sample have worked on product systems during their careers, as illustrated in Figure 30; less than 20% have worked on service systems, and about 25% have worked on enterprise systems. Only senior systems engineers reported having responsibilities at the system of systems level. Because most of the experiences discussed during Helix interviews were on product systems, no insights can be highlighted that focus specifically on service or enterprise systems. However, it was indicated that working on service or enterprise systems required a much richer understanding of business in general – business processes, finance, and the overall goals and drivers of that particular business. It was an area where educational elements such as an MBA might be very useful.

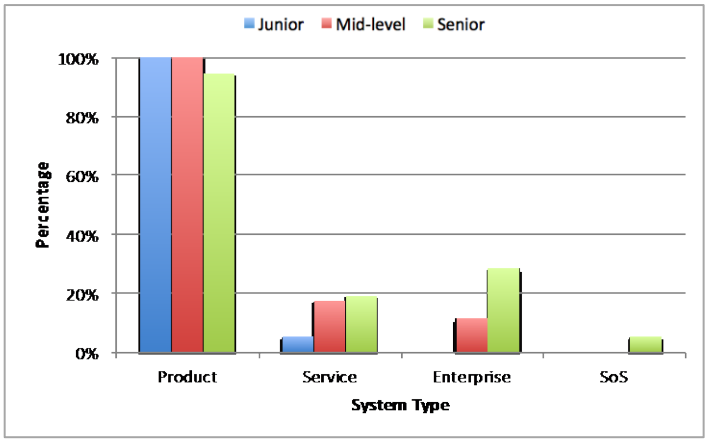


Figure 30. Experiences of Systems Engineers across System Types, by Seniority

The definitions for product, service, and enterprise align with the definitions found in the SEBoK.

- **Product** – A system considered from the point of view of a physical "system end product" (ANSI/EIA 2003) made of system elements that may include hardware, software, infrastructure and support services. The people and organizational aspects of the "whole system" of which the "product system" forms a part have to be considered in the design, but are provided by another organization.
- **Service** – A dynamic configuration of resources (people, technology, organizations and shared information) that creates and delivers value between the provider and the customer through services.
- **Enterprise** – A complex, (adaptive) socio-technical system that comprises interdependent resources of people, processes, information, and technology that must interact with each other and their environment in support of a common mission to offer an output such as a product or service.

System Scope

Identification of system scope tends to be relative, depending on the size and complexity of the system, and on the system level at which the organization is engaged. Based on what could be identified from the Helix sample, Figure 31 shows the experiences across systems levels, divided by seniority. The definitions of the different system scopes are taken from the SEBoK:

- **Component** – An entity with discrete structure, such as an assembly or software module, within a system considered at a particular level of analysis.
- **Subsystem** – A self-contained system within a larger system.

- **System** – A self-contained combination of interacting components organized to achieve one or more stated purposes.
- **Platform/System of Systems** – Two or more systems that are separately defined but operate together to perform a common goal.

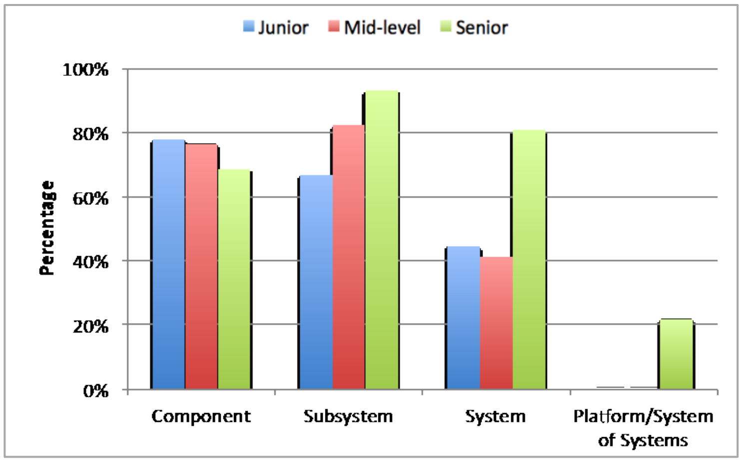


Figure 31. Experiences of Systems Engineers across System Scope, by Seniority

The following observations can be made from Figure 31:

- Not all systems engineers have worked scope of dealing with an entire system. However, many of the systems discussed during interviews were large and complex. When a systems engineer worked on subsystems, it could be complex enough to be considered the "system" by a different organization, but was still counted a 'subsystem' level experience based on the language used by the interviewees.
- Junior and mid-level systems engineers make up almost a 1/3 of the sample; and so, some of them have not yet had responsibilities at the whole-system level.
- Not all individuals have had responsibilities at the component level. This may be a result of what an individual reported in the resume or interviews versus what their full range of experiences have included.

Experiences at different system scopes were generally linked with development of different types of proficiencies.

- Experiences at the **component level** were generally associated with development of proficiencies in *Math/Science/General Engineering*, as these were often experiences applying engineering disciplines at the detailed level. In addition, it was common for early leadership positions to be at a small scale – component or perhaps a small subsystem – and these experiences helped to develop proficiencies in *Technical Leadership*.
- Experiences at **subsystems level** often created opportunities for engineers to better understand the implications of the environment or concept of operations on the operation of the system, helping them develop higher proficiencies in *System's Domain and Operational Context*. Because subsystems require the interface and interaction of multiple components, they also provide more opportunities for understanding and growing proficiencies in *Systems Engineering Discipline* and *Systems Engineering Mindset*.

- At the whole-**system level**, systems engineers were generally taking on higher levels of responsibility and working on larger teams – further growing their proficiencies in *Interpersonal Skills* and *Technical Leadership* while also requiring them to further grow in their understanding of *Systems Engineering Discipline* and the application of the *System's Domain and Operational Context*. Systems engineers often stated that a higher proficiency in *Systems Engineering Mindset* was required in order to be effective at this level.

Experiences Over Time

The data presented in Section 4.8.6, above, provides a cumulative view of the experiences across the entire Helix sample. However, experiences become most interesting when patterns are assessed over time. This data is captured and presented in Sections 5.1.1 - 5.1.3, below.

4.5.2 FORCE 2: MENTORING

Mentoring (or mentorship) is a relationship between two individuals: a mentor possesses more experience and knowledge and shares these with a mentee for the mentee's personal development. The effectiveness and derived value of the mentoring relationship is dependent on the individuals involved, but is also influenced by the organization which derives value out of a mentoring relationship as well.

What is Mentoring?

Mentoring means different things to different individuals and in different organizations. Common characteristics of mentoring are discussed below.

- Two individuals are involved in a mentoring arrangement: a mentor and a mentee (also referred to as a protégé).
- The mentor is usually *senior* when compared to the mentee in age, experience, and/or expertise.
- Primarily, the mentor *gives* and the mentee *receives*.
- The mentor-mentee relationship is a many-many relationship: a single mentor can have multiple mentees, and a single mentee can have multiple mentors – concurrently or spread over time.
- Mentor-mentee interactions typically happen over an extended period of time at varying frequencies.

There are also some differences and contradictions in the understanding of mentoring.

- Some use the term mentoring to describe any interaction with any co-worker in the organization that would provide any advice or guidance to handle the problem at hand.
- Some consider mentors to be synonymous with subject matter experts (SMEs) who are consulted for their expertise on an as-needed basis only. In contrast, some consider it mentoring only if the mentor is a senior person, and only if there are regular interactions between the mentor and mentee over an extended period of time.
- When the mentor and the mentee are of the same seniority in terms of age, years of experience, or level of expertise, some still consider it to be a mentoring relationship, while some others consider it to be a peer-peer relationship and not a mentoring relationship.
- Some distinguish between the concepts of coaching and mentoring: coaching is related to providing advice and guidance on solving a specific technical problem, while mentoring on the other hand, has neither a set beginning or end to the relationship nor is related to a specific event.

Mentoring Arrangements

Mentoring arrangements can either be formal or informal, depending on the level of engagement of the organization in establishing and sustaining the mentoring relationship. The two types of mentoring arrangements may be summarized as below:

- **Formal**: The organization plays an active role in establishing the mentor-mentee relationship, and also lays down guidelines for maintaining that relationship. Usually, organizations require that objectives and expectations for the mentor and the mentee be stated explicitly. The relationship and its progress tend to be monitored by the organization.
- **Informal**: The participating individuals establish the mentor–mentee relationship by themselves:

either a mentor adopts a mentee or a mentee seeks out a mentor, and the relationship is established. Formal objectives or expectations are usually not stated explicitly, but it is considered good practice to establish these in some form at the start of the relationship. The organization plays a less active role in informal mentoring. It is upon the mentor and the mentee to establish and drive the relationship.

Mentoring Focus

Depending on what the mentoring is about, interviewees mentioned three types of mentoring:

- **Career Mentoring:** The mentor provides advice on career-related issues: helps identify career goals and the paths leading to that goal. The mentor could be from another group or division in the organization. Mentees are also groomed on management and leadership related topics.
- **Technical Mentoring:** The mentor typically provides advice on the technical details of the system being engineered. The mentor teaches lessons that are typically not found in textbooks and provides crucial insights on technical tools and processes. The mentor also acts as a subject matter expert, answering questions mentees might have on the subject, the system, or the program.
- **Organizational Mentoring:** While closely related to career mentoring, in organizational the mentor provides information about the organization: its culture, its procedures, and its policies. This is especially critical to a new employee.

Analysis of Mentoring

The topic of mentoring was discussed with 120 Helix interviewees; and among them, 80 discussed the mentoring that they had received or provided during the careers.

Engagement in Mentoring Arrangements

Figure 32 indicates the type of mentoring received by interviewees anytime in their career. Among the 80 interviewees used for this analysis, 89% had received some type of mentoring (11% did not receive any type of mentoring). Within this, 12% had received formal mentoring only and 36% had received informal mentoring only. Among the 23% who had received both formal and informal mentoring, informal mentoring was more valuable than the formal mentoring they had received.

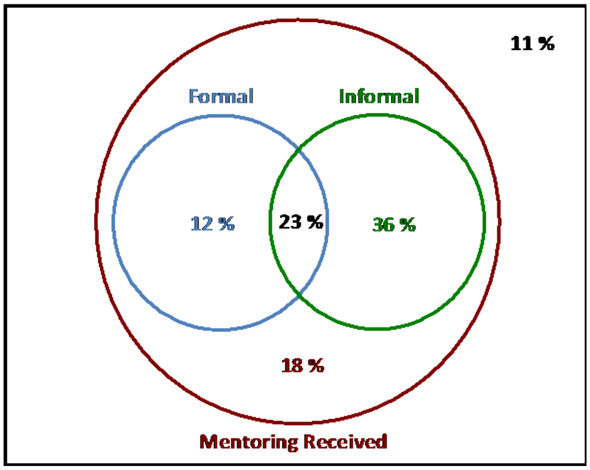


Figure 32: Types of Mentoring Received by Interviewees

Figure 33 indicates the mentoring arrangements where interviewees were the mentors. Most of the 80 interviewees had indicated that they had been mentors at some point in their careers, and many were active mentors during the time of the interview. It was rare to find an interviewee who was a mentor in the past but was not mentoring anyone at present, and those who had found much value in the mentoring they had received were eager and willing to mentor others. This indicates that interviewees cease to be mentees at some point in their careers, but rarely cease to be mentors. Among 80 interviewees used for this analysis, only 10% had not mentored anyone. In the 90% who have provided some form of mentoring to others, 4% had been a formal mentor only and 51% had been informal mentors only, but 15% had been both formal and informal mentors. Interviewees preferred to mentor others, especially those junior to them, in an informal arrangement rather than in a formal arrangement set up by the organization.

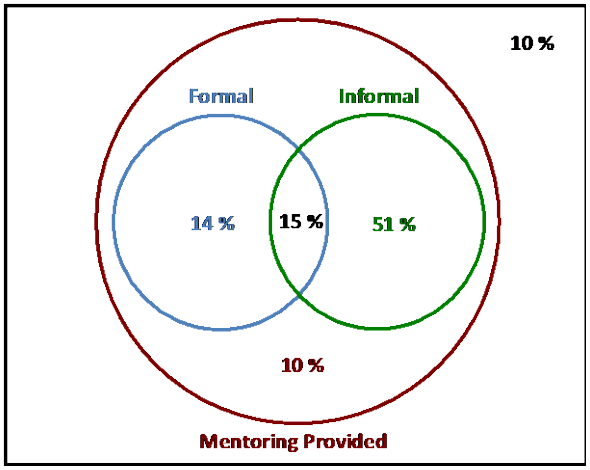


Figure 33: Types of Mentoring Provided by Interviewees

Formal vs. Informal Mentoring

In all the mentoring related discussions, the biggest debate was on which mentoring arrangement was more effective: formal or informal. Both these forms of mentoring have their own pros and cons; and in many cases, it appeared that the challenges and limitations in one type of mentoring were easily resolved by the other. Though the interviewees expressed their strong preferences, there was no clear winner, nor was there meant to be one. Either of these could be effective, depending on the organizational policies and culture, or depending on the specific mentor and mentee involved in the relationship. In any case, both formal and informal mentoring arrangements have their place in any organization and in any mentor-mentee relationship. As one interviewee put it, “its not about formal or informal as long as there is a commitment from me, the mentor, to spend the time to answer the questions, and commitment from [the mentee] to put some effort in”. The views expressed by the participants are tabulated below.

Table 9. Comparison between Aspects of Formal and Informal Mentoring

Topic	Formal Mentoring	Informal Mentoring
Visibility to Organization	Organizations tend to have full visibility into the mentoring arrangement and how it is working out for the mentor and the mentee.	Organizations are usually unable to keep track of a mentoring arrangement, and may even be unaware that such arrangements exist.
Mentor-Mentee Pairing	Organizations enable the mentor-mentee pairing: in some cases, it is forced; and in some cases, some flexibility in choice is given. There is the possibility of “wrong” selection or pairing that may not last long. In some cases, organizations enable mentors and mentees to establish a mentoring arrangement, but do not explicitly make the pairing.	Mentors and mentees tend to establish the relationship by themselves, usually upon the request of the mentee. These relationships tend to last longer, since the mentees have the flexibility to choose the mentor that they are comfortable with.
Mentor Engagement	Not all senior engineers are good mentors; and not all potentially good mentors are willing to be one. If they are forced into a relationship against their preference, they tend to be ineffective and unwilling. “Some mentors don’t interact at all,” said one interviewee.	Mentors usually enter into a mentoring relationship by their own choice and therefore tend to be more engaged.
Mentee Responsibility	The organization plays a part in establishing a mentoring arrangement, and this works for introverted mentees. However, it could also make mentees more passive than active.	The mentees must find a mentor and drive the relationship; “I think the mentee has to want it more than the mentor,” said one interviewee. More introverted mentees may find it difficult to seek a mentor and to ask them questions, while extroverted mentees find it easier to “go bug them and pick their brains”.
Goals and Objectives	The organization lays down expectations for a mentoring relationship and could also provide guidance on establishing goals and objectives. This is helpful to the mentee in particular, and tends to be more impactful.	There may be an informal understanding of some overall goals and objectives between the mentor and the mentee, but there is no requirement to establish them.

Topic	Formal Mentoring	Informal Mentoring
Mentoring Load	Mentoring can be burdensome, mechanical, and obligating. It is possible to "go through the motions" without any beneficial engagement.	Mentoring is usually a pleasure for both the mentor and the mentee since they both tend to be "willing and eager". "If you try to formalize or institute a mentoring program, it feels awkward" said one interviewee.

Benefits of Mentoring

In any typical mentoring arrangement, the mentor 'gives' and the mentee 'receives'. Therefore, such an arrangement is expected to be most beneficial to the mentee. However, there are benefits to the mentors as well. In addition, the organization also stands to benefit. Whenever an organization establishes a formal mentoring initiative, it usually expects to derive some benefit from the mentoring arrangements. However, the benefits to the mentee, to the mentor, and to the organization are conditional, and may not be taken for granted. The discussions presented in this section also establish the need for mentoring.

Benefits to Mentees

A mentee is often a new or younger employee of the organization. Although mentees may have formal education or training that qualifies them for their jobs, there is much to be learned in the context of the organizational environment and culture and in the nature of the systems being engineered. A mentee could also be a senior engineer who is either new to the organization or has moved from another part of the organization, and whose experience and expertise could be in a different discipline or system. In either case, the mentee gains significantly through mentoring:

- **Relationship with Mentor**: The biggest benefit to mentees of mentoring is the relationship they establish with their mentors over the span of their careers; most other benefits of mentoring are enabled through the mentor. Mentors become key enablers for mentees: they look after them; they are their biggest advocates and champions; they help identify strengths; and they become critical factors for success in their careers. On the professional achievements he has accomplished, one interviewee said, "Without my mentor, it might not have happened." Not everyone is lucky to have a mentor early in their career, and some interviewees stated that their careers could have been different if they had had a mentor. When asked what he would do differently if he were to re-start his career, one interviewee answered, "I would get a mentor."
- **Increased Effectiveness**: Most interviewees identified mentoring as a critical factor that increases the effectiveness of systems engineers. The lessons that they learn and knowledge that they acquire through mentoring is less effectively attained through other means. Interviewees also noted that through mentoring, the learning is quicker and more effective when compared to other means.
- **Career Advancement**: Through their mentors, employees often get exposed to opportunities within the organization that may not be visible otherwise. Mentors tend to point the mentees in the 'right direction' and enable them to move in that direction. In hindsight, such moves have been significant contributors for the career advancement of many interviewees.
- **Valuable Lessons**: During mentoring, mentees often receive important lessons from their

mentors, which have made a significant impact in their careers. Many interviewees quoted their mentors during the interviews; and how something they learned from their mentors shaped their perspectives. Mentees also learn how to cope with challenges, how to deal with complexity, and also receive tactical guidance from Mentors on “how to do this job well.” Mentees also acquire critical insights into the system or program that help them over their career.

- **Strong Networking**: Building a strong professional network is key to any employee. Through mentoring, mentees get exposed to many senior engineers and others in the professional network of their mentors. In addition, mentors could also rise to very senior positions within the organization and continue to contribute to the network of the mentee.

Benefits to Mentors

A mentor is usually a senior, experienced engineer who has spent more time in that part of the organization into which the mentee has entered, or in dealing with the type of system that may be unfamiliar to the mentee. Though the mentee stands to benefit the most, the mentor also benefits from mentoring which tend to motivate the mentor to engage in a mentoring a relationship.

- **Professional Gratification**: “If I can get people to do better at what they’re doing, that’s an incentive for me to stick around” said one interviewee who was close to retirement; “I mentor people just because I want to,” said another. Many considered mentoring to be an important part of their jobs, irrespective of the organizational acknowledgement of it. “Helping the stars” and “teaching the young’uns what to do” seems to be motivation enough for the mentors. One interviewee even said about those he mentors, “their work portrays my effectiveness”.
- **Organizational Recognition**: In organizations where mentoring is acknowledged, mentors get recognized for their efforts. It is also typical for mentoring to be featured in the annual performance reviews of the mentors. Being a mentor increases the visibility of an employee within the organization and also helps in career advancement.
- **Reduced Workload**: Some mentors considered mentoring to be a means of reducing their workload. When a mentee is able to share the load, “I don't have to work as many hours,” said one interviewee.
- **Grooming Successor**: An interviewee said that an important lesson that his mentor had taught him was that “you can’t advance [in your career] without training a person to replace you” and that by following that advise, he has been presented with lots more opportunities than his peers. “Somebody’s got to take over” the mentor’s position in the organization – not just after retirement or after moving on to a different job, but also in the absence of the mentor. One interviewee who occupied a senior position in the organization had met with an accident and had to stay away from work for a few months during recovery. But since he had previously mentored someone, that person was able to fill the gap during his absence on the job.

Benefits to The Organization

Effective mentoring not only benefits the mentees and mentors involved in the relationship, but also the workforce as a whole. When this happens, the organization at large benefits as well. One of the early motivations of the Helix study was the concern that a significant number of senior systems engineers would soon be eligible for retirement and whether the organization would be able to effectively manage the gap that would be created. Mentoring is considered to be an effective way to help address this gap.

- **Gain Effective Knowledge Transfer**: When senior engineers retire, they typically take along with them many years' worth of valuable experience and expertise, and this is a major concern for any organization. Among various knowledge transfer mechanisms that organizations typically deploy, interviewees indicated that mentoring is very effective. In a couple of scenarios, organizations had budgeted time for documentation, but interviewees agreed that it was not a good decision and that mentoring would have been more effective. In one organization, interviewees felt that the organization was effectively handling the workforce situation (of retiring senior engineers) by including mentoring as part of the transition plan in the workforce.

- **Identify High-Potential Engineers**: Through the feedback from mentors, organizations can identify high-potential engineers who are being mentored. These young engineers could then be placed on a fast track or experience other organizational initiatives that would mature them faster into the next generation of experts and leaders.
- **Reduce Orientation Time**: When employees enter a new organization or department, irrespective of their educational and work background and experience, it takes some effort by the organization and by the individual before they can become productive and effective in their new roles. Most often, individuals spend a lot of time figuring out things by themselves, and in searching for the information and organizational procedures required to do their jobs. In these cases, effective mentoring can significantly reduce the time taken for such employees to get oriented to their jobs.
- **Fill Workforce Gaps**: When senior engineers retire, mid-level engineers most effectively fill the gap created and in many organizations there are not enough mid-level engineers. Through mentoring, such emerging workforce gaps can be filled proactively. In other situations, there exist some 'less sought after' jobs in an organization that are still important, e.g., test engineer. While talking about his job, one interviewee mentioned, "you would be surprised how many people don't want to do this job!" This could be due to a combination of organizational culture or misconception about what that job entails. Again, mentoring is an effective way to fill such workforce gaps that may not be filled easily otherwise.
- **Increase Employee Retention**: "Today, employees do not stay on with companies as long as they used to," said one interviewee. When younger employees leave, they also take along with them the learning and knowledge that they have acquired, and the organization stands to lose. When employees are recognized by the organization for their potential; when the organization and seniors invest time and effort on them; and when their work is deemed important, it significantly increases the loyalty of those employees towards their job and the organization. Mentoring helps increase such employee loyalty and thereby employee retention.
- **Improve Organization Culture**: Mentoring builds relationships and thereby naturally builds teams. It promotes a healthy environment where senior engineers are willing to share and teach, and junior engineers are recognized and encouraged. This creates a positive organizational culture that all employees are happy to be a part of. In one organization, "there a number of people that are willing to help somebody – not 10% but 90% of the workforce; it is just part of the culture" said one employee.

Importance of Mentoring for Systems Engineers

Mentoring, in general, is helpful to any mentee in any organization – and not just for engineers. However, in the context of systems engineers and systems engineering, mentoring plays a particularly important role.

- **Systems Engineers Can be a Rare Commodity**: In some organizations, there are not enough systems engineers to perform the required systems engineering activities. In such cases, mentoring is an effective way to fill the pipeline of systems engineers, especially when there is likely to be a gap with the retirement of senior systems engineers. “There are not [as] many systems engineers as I thought,” said one interviewee, “without mentoring there will be a loss”.
- **Identifying and Recruiting Systems Engineering talent**: In many organizations recruitment directly into the systems engineering division or into a systems engineer’s role does not happen; systems engineers tend to be brought in from other parts of the organization. In these cases, mentoring plays a crucial role where mentors help identify potential systems engineers and play the role of an advocate for systems engineering to encourage them to join systems engineering.
- **Support for New Systems Engineers**: As one participant said, “It isn’t rare and it isn’t uncommon to end up doing systems engineering if you were not a systems engineer before”. In most organizations, this is a common way for recruiting systems engineers. So when non-systems engineers enter systems engineering, mentoring plays a key role in equipping them to be effective.
- **Changing Face of Systems Engineers**: Systems engineers used to be ‘greybeards’ who “floated up to the top and had all the experience”. But today, depending on organizational policies and practices, engineers may become systems engineers without a lot of experiences. Mentoring becomes a critical initiative that could equip such systems engineers to be effective in their jobs.
- **Nature of Systems Engineering:** “As much as we like the young engineers to act as systems engineers, it’s difficult for them to understand how our different taskings are integrated across,” said one interviewee. “There’s a big gap between young engineers and old experienced people with a lot of tribal knowledge” said another. Due to the nature of systems engineering, and particularly how it is performed in the organization, there is much to be learned hands-on that cannot be learned before entering the organization. Though education does help, it is not sufficient in most cases. One interviewee elaborated, “I had the advantage as a teenager, some of my mentors had been doing the work since the 1950’s and they were about ready to retire. They were solving incredibly complex problems without computers, without high-end sensors. They had to really understand what they were doing and really have clever ways of solving the problem. So they really kind of passed on that skill set that you don’t learn in college.”

Critical Factors for Success with Mentoring

Though mentoring is considered to be an effective solution to address the gap that could be created by an aging retiring workforce, just the existence of a mentoring arrangement, either formal or informal, by itself does not guarantee success. However, there are factors that significantly influence the success or failure of any mentoring arrangement, and interviewees provided a number of anecdotes and personal experiences to elaborate on this. Recognizing and addressing these factors by the organization and the individuals involved could prevent failure and could vastly increase the chances of success with a mentoring arrangement.

- **Mentoring Has Its Limitations**: Mentoring does not always work in all organizations, for all employees; mentoring does not work in isolation, but in conjunction with other organizational initiatives to support its employees, such as training and education; and mentoring cannot be solely relied on for knowledge capture from a retiring senior engineer. Even with respect to mentees who stand to benefit the most in a mentoring arrangement, one interviewee stated, “I do not see different traits among those formally mentored, informally mentored, and not mentored at all." While there are many other cases that would prove this statement wrong, it only shows that mentoring is not a universal solution. One interviewee stated "I've seen great systems people that have no formal mentoring". While it is possible that any mentoring could have been implicit and not visible outside, it shows that while mentoring is highly desired, it may not be essential for everyone. Another interviewee supported this view when he said "I'm proof that [mentoring] is not essential; it just made it more uncomfortable for me."

- **Right Choice, Right Pairing**: Ineffective mentors exist, and just the fact that someone is a true expert with many years of experience would not automatically make them a great mentor. In some mentors 'communication' could be a factor, but 'willingness' could also be a factor. As one interviewee elaborated, "some are good with sharing their knowledge, and there are people who don't want to share any of their knowledge". Similarly, not everyone benefits from being a mentee; some are more comfortable learning things on their own than from a mentor. Finally, it is most important to pair the right mentor with the right mentee – alignment between them is important; "[mentoring] is a two way street" said one interviewee, and it can become a "life long relationship".
- **Balance between Formal and Informal Mentoring**: While most interviewees had a personal preference for informal mentoring, everyone agreed that a bit of formality does help. Therefore, establishing the right balance between formal and informal mentoring by the organization is critical. As discussed in Section 0, there are some aspects of mentoring that benefit from a formal arrangement – such as establishing goals and objectives, and keeping track of mentoring arrangements. But for aspects such as deciding the right mentor-mentee pair; and for driving the frequency and nature of the interactions, it is advantageous to let those be informal.
- **Mentor Training**: Mentors may possess the experience and expertise that mentees stand to benefit from, but some guidance on establishing and sustaining a mentoring relationship would help – it does not come naturally to all mentors. Providing some guidance on how to establish goals and objectives for mentoring, and on how to balance teaching and guiding (for self learning by mentees) would be beneficial to mentors. In one organization, interviewees mentioned that there was a mentoring manual, but it was very dated, and that currently mentoring happens in a more ad-hoc manner.
- **Make Mentoring Visible**: If any mentoring program is officially established or encouraged by an organization, it must also take the effort to make it known to all its employees. In some organizations, the interviewees were not aware if their organization had a mentoring program or not. In some cases, top management initiated a mentoring program that failed to percolate through the organization to reach the potential mentors and mentees. One interviewee said, "I didn't even know we had [a mentoring initiative] until someone gave me a mentee."
- **Mentoring Needs Time**: Benefits of mentoring cannot be reaped instantaneously – in some cases, it is only after a number of years that mentees benefit from the mentoring that they had received early on in their careers. Similarly, mentoring cannot be a last minute activity that a retiring senior engineer is expected to do in the last few months or weeks of employment – any mentoring done at that time is not likely to be very effective. With many other activities taking priority before retirement, there is usually not enough time mentoring. Mentoring cannot be rushed, even from the perspective of the mentee who cannot be expected to "think of all questions [to the mentor] today."
- **Load on Mentor**: Organizations must always be aware of the load that mentoring places on the

mentor – they cannot be expected to do everything that they are responsible for and also do mentoring. At the same time, mentoring cannot be a fill-in activity where a mentor is told, “you are not busy this month – so go mentor this person.”

- **Back up a Mentoring Program**: In one organization, a formal mentoring program was rolled out where everyone would have a mentor and a mentee. It was formal; but “it wasn't backed by a lot of horse power”, as noted by an interviewee. The organization needs to back up a mentoring program with the required budget and time for the mentor and mentee to engage in a relationship – it is dangerous to establish a mentoring program just to claim that one exists. Breaking a mentor-mentee relationship can also be harmful – such break ups could happen for a number of reasons, but if it is something that an organization could prevent, it should.
- **Terminating a Mentoring Arrangement**: Even while a mentoring initiative continues to remain active, there are situations that may warrant the termination of a mentoring arrangement. When the mentee chooses to leave the organization, it could be a loss to the organization if the mentee had received mentoring from soon-to-be-retiring senior mentor. Similarly, when a senior mentor chooses to leave abruptly, it could affect the knowledge capture that may have been in process through the mentoring arrangement. However in either of these cases, the organization may not be able to do much. But when there is a likelihood of an existing mentoring arrangement to be disrupted due to promotions or internal re-organization of either the mentee or the mentor, the organization must take care to provide sufficient time to the mentor and mentee to terminate the mentoring arrangement in a cordial manner, especially if the mentoring arrangement was successful and valuable to the mentor and the mentee.

4.5.3 FORCE 3: EDUCATION & TRAINING

Education plays two key roles in the development of systems engineers:

1. It provides the foundation knowledge to support engineering-related work. Typically, this takes the form of undergraduate education in an engineering discipline, technical field, or physical science.
2. Graduate level education is an avenue to develop more advanced skills, explore more in-depth knowledge, and help systems engineers grow as they move through their careers.

In addition to formal academic programs leading to undergraduate and graduate degrees, there are graduate certificates that individuals obtain, in an area that is closely related to their work. Some systems engineers go on to obtain doctoral degrees as well.

Systems engineers typically start their careers after obtaining an undergraduate degree, while graduate degrees may be obtained immediately after an undergraduate program or after a few years of professional work. Any formal degree directly improves proficiency in the relevant areas and categories. Any undergraduate degree in engineering typically provides much of the *Math/Science/General Engineering* proficiency in addition to the relevant categories under the *Systems' Domain & Operational Context* proficiency area. Graduate degrees add to relevant proficiencies; much of the formal systems engineering education happens at the graduate level.

While academic programs are typically offered by a university, there are a number of tailored training programs that organizations offer their employees. These trainings are more focused on building specific skills that are required for them to perform their work and are typically offered short-term. The topics vary widely across organizations, with some training focused on the technical aspects of systems development, other training focused on organization-specific approaches and processes, and still other training focused on leadership or interpersonal skills. Each type of training has a role in the development of proficiency.

Among the six proficiency areas in *Atlas*, *Math/Science/General Engineering*, *System's Domain & Operational Context*, and *Systems Engineering Discipline* may be considered to be 'hard' proficiencies at large, while *Systems Engineering Mindset*, *Interpersonal Skills*, and *Technical Leadership* may be considered to be 'soft' proficiencies at large. Formal education typically improves the hard proficiencies, but training could improve both hard and soft proficiencies.

In general, education or training results in an initial, single increase in proficiency. Additional changes over time are then the result of applying the knowledge or skills gained through this force in a real-world setting; i.e., through experiences utilizing the outputs of the education or training.

Analysis of Education

While the third force identified in *Atlas* is Education & Training, training is very specific to the organization, and there was insufficient and inconsistent information to perform any further analysis on training. Therefore, only the education of systems engineers, gathered primarily from their resumes, is included here. A comparison between junior systems and senior systems engineers provides additional insights into early career development of these two seniority levels. Data from INCOSE SEP applicants and analysis performed on their education background (Lipizzi et. al 2015) has also been used here.

Degrees Awarded

There are many types of degrees awarded to individuals in the interview sample. For Helix analysis, degrees were categorized by level: associate, bachelor's, master's, and doctorate degrees. For the purposes of this analysis, graduate certificates were not included.

The highest degree attained for systems engineers in the Helix interview data, divided by seniority, as well as applicants of the INCOSE SEP program, are listed in Table 10. Although there is very little overlap between the interview participants and the INCOSE SEP applicants, the distributions across degrees is fairly well aligned in both samples.

Table 10. Highest Degree Attained by Individuals

	Helix Interview Data				INCOSE SEP Data
Degree Level	Junior	Mid-level	Senior	All	
Associate's	0%	0%	0%	0%	<1%
Bachelor's	44%	23%	32%	33%	30%
Master's	56%	73%	56%	58%	61%
Doctorate	0%	5%	12%	9%	8%

The following observations can be made from Table 10:

- All individuals in the interview sample have at least a bachelor's degree; very few of the INCOSE SEP applicants have only an associate's degree.
- The majority of the interviewees have a master's degree (58%), which is close to the fraction of INCOSE SEP applicants (61%). In general, systems engineers are a highly educated group.
- Mid-level systems engineers in the interview sample have a much higher percentage of master's degree attainment than either junior or senior systems engineers. It is possible that this is reflection of the distribution of the sample population, by seniority. Another plausible explanation is that many junior systems engineers haven't been practicing professionals long enough to seek a master's degree and that many senior systems engineers grew up in a time when advanced degrees, particularly in systems engineering, were not as important for career advancement as they are now.
- Among interviewees, there is a higher percentage of senior systems engineers with PhDs than mid-level systems engineers; no junior systems engineers in the sample have a doctorate degree.

Bachelor's Degrees

Bachelor's degrees generally provide the foundation on which a systems engineer's career is built. Figure 34 shows the most common majors in bachelor's degrees among Helix interviewees and INCOSE SEP applicants. There is a wide variety of degree titles, especially in the INCOSE SEP data that includes non-US applicants. In the interview sample, there were 81 total degree titles; in the applicant sample, there were 453 unique degree titles for bachelor's degrees and 524 for master's degrees. All these titles were normalized into the major categories used in Figure 34 (Lipizzi et. al 2015).

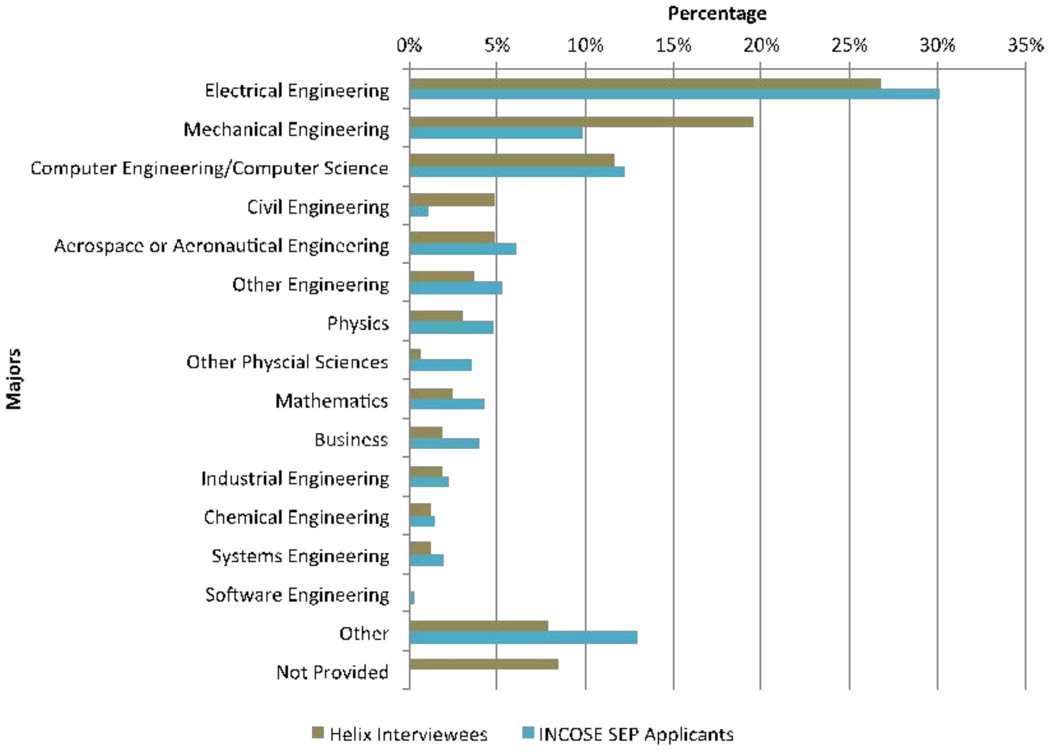


Figure 34. Bachelor's Degree Majors for Helix Interviewees and INCOSE SEP Applicants

The following observations can be made from Figure 34:

Though the order and percentages differ slightly between Helix interviewees and INCOSE SEP applicants, the overall pattern in popularity of these majors is generally aligned in both samples.

- Among interviewees, over 75% have bachelor's degrees in engineering fields; a small number were educated in the physical sciences (4%) or mathematics (2%).
- Bachelor's degrees in systems engineering (and related degrees) are rare, at just over 1%.
- The majors represented under "other" were not technical fields (e.g., French, music, quality assurance, and leadership). A few individuals had more than one bachelor's degree, but this was rare (2%) and typically the second degree was not technical (e.g., music and French).
- In both data samples, the most common bachelor's major is electrical engineering. In the 1970's, electrical engineering was a more common major in the US than mechanical; this trend has reversed over time (NCES 2010).

Bachelor's degrees were attained at the start of or within the first 3 years of all systems engineers' careers in the interview sample. Table 11 shows a comparison of bachelor's degree majors for junior versus senior systems engineers.

Table 11. Comparison of Bachelor's Degree Majors of Junior and Senior Systems Engineers

Bachelor's Degree Majors	Junior	Senior
Electrical Engineering	26%	34%
Mechanical Engineering	26%	22%
Civil Engineering	15%	2%
Computer Engineering/Science	15%	9%
Aeronautical/Aerospace Engineering	7%	6%
Systems Engineering	0%	1%
Software Engineering	0%	0%
Engineering, other	7%	1%
Physics	0%	6%
Other Physical Sciences	0%	1%
Mathematics	0%	3%
Business	4%	1%
Other	0.%	13%

The following observations can be made from Table 11:

- When many senior systems engineers were working on their bachelor's degrees, electrical engineering was a more popular major.
- Civil engineering and computer engineering/science are more common in the junior systems engineering population than in the senior.
- None of the junior systems engineers in the sample have undergraduate degrees in physics, other physical sciences, or mathematics; these are present, though uncommon, in the senior systems engineer population. Among interviewees, individuals with degrees in these areas typically began their careers conducting analysis, conducting tests, working on requirements gathering, or in planning. These individuals did not conduct detailed analysis or larger-scale verification and validation efforts.

In the interview sample, three paths were observed:

1. Individuals started in a non-technical field but later sought a bachelor's degree in a technical field;
2. Individuals had one or more minors in technical fields, and utilized those for their career growth;
3. Individuals came to engineering based on a circuitous route and first gained engineering experience before pursuing formal education in these areas.

Master's and Ph.D. Degrees

Over 50% of the Helix interviewees have a master's degree, as well as almost 2/3 of the INCOSE SEP applicants. Of the Helix interviewees with master's degrees, 12% earned more than one master's degree.

These percentages are considerably higher than in the general US population, in which around a 1/3 of individuals have a bachelor's degree and around 1/10 have graduate level education (US Census Bureau 2014). This indicates that systems engineers, as a group, are much more highly educated than the general population.

The majors for master's degrees for both the Helix interviewees and the INCOSE SEP applicants are shown in Figure 35:

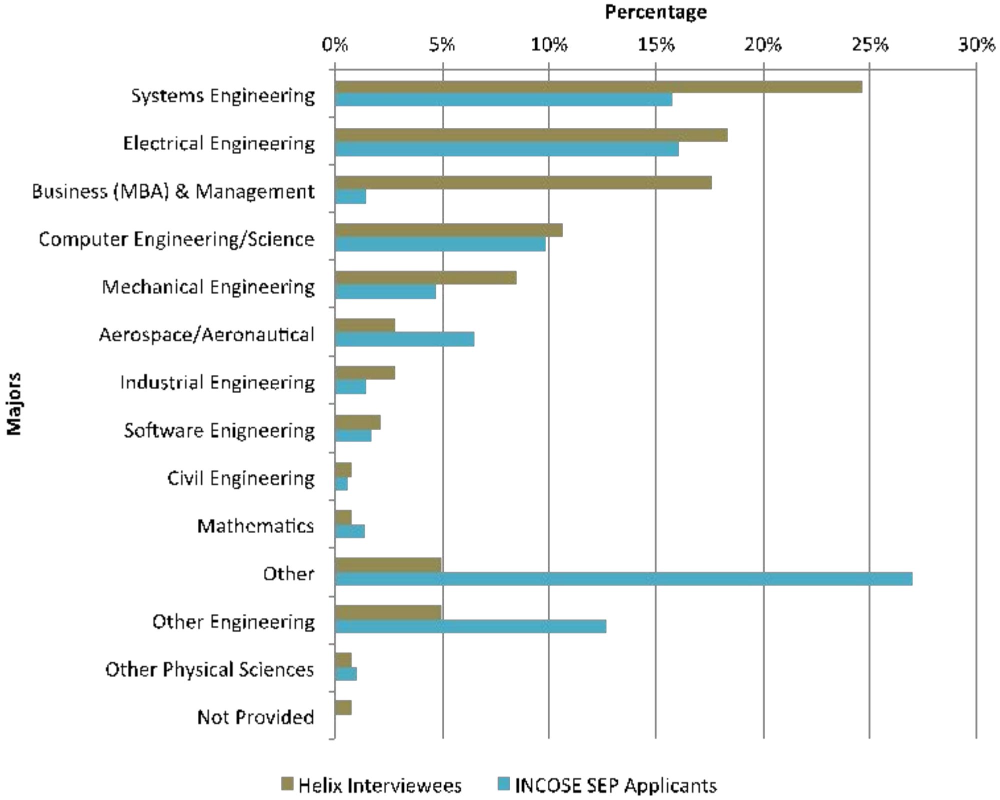


Figure 35. Master's Degree Majors for The Helix Interview Data and INCOSE SEP Data

The following observations can be made from Figure 35:

- Almost 3/4 of the Helix interviewees and over 2/3 of the INCOSE SEP applicants have a master's degree in an engineering field.
- Both samples have very few individuals (less than 1% in each) with master's degrees in the physical sciences.
- The most common master's major among interviewees and one of the most common in the INCOSE data was systems engineering. A quarter (25%) of the Helix interviewees and 16% of the INCOSE SEP applicants have a master's degree in systems engineering.
- The high percentage of 'Other' in INCOSE data indicates that the majors, especially from non-US universities, did not directly fall under any other category.

Table 12 shows a comparison of master's degree majors for junior versus senior systems engineers in the interview sample.

Table 12. Comparison of Master's and PhD Degree Majors of Junior and Senior Systems Engineers

Degree Majors	Master's		Ph.D.
	Junior	Senior	(Senior)
Electrical Engineering	10%	21%	17%
Systems Engineering	50%	20%	33%
Computer Engineering/Science	5%	11%	0%
Mechanical Engineering	5%	10%	8%
Aeronautical/Aerospace Engineering	5%	3%	0%
Software Engineering	5%	1%	0%
Civil Engineering	5%	0%	0%
Engineering, other	0%	9%	8%
Physics	0%	1%	8%
Physical Science, other	0%	1%	8%
Mathematics	0%	1%	0%
Management	0%	18%	8%
Other	5%	4%	8%

The following observations can be made from Table 12:

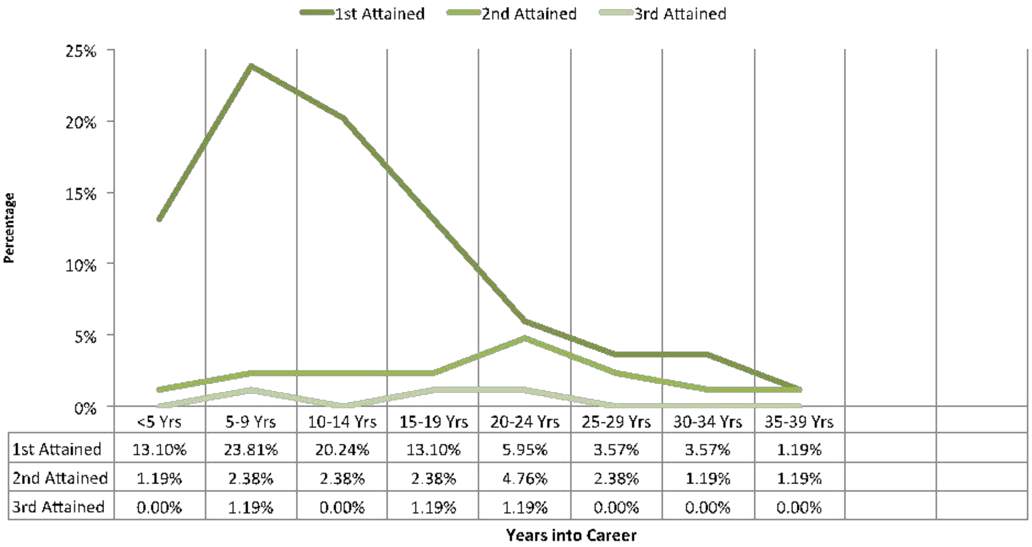
- As indicated in Table 10, no junior systems engineers had a Ph.D. degree in the Helix interview data.
- In both master's and PhD degrees, systems engineering and electrical engineering are the most common fields of study.
- While 18% of master's degrees attained were in business or management fields in the interview set, only 6% of the INCOSE SEP applicants have business or management-related master's degrees. Among INCOSE SEP applicants, 41% of individuals with a master's of business degree also have a technical degree.
- Among interviewees, most PhD majors were unique with two exceptions: systems engineering (33%) and electrical engineering (17%).

Research by Lasfer and Pyster (2013) shows very strong growth in systems engineering master's graduates from 2001-2010. The increased popularity and availability of systems engineering programs in the 2000's then, is common throughout the systems engineering community in the US. Graduate education in systems engineering has grown significantly in the last 15 years, and senior systems engineers have proven less likely to pursue this degree. Growth of academic programs generally occurs in response to industry demand. Almost all interviewees who were asked about the growth of systems engineering in their organizations explained that it has only been in the last 15-20 years that the field of systems engineering has come to be clearly defined as a separate discipline and is recognized and pursued by participating organizations. Industry interviewees frequently cited increased governmental focus on systems engineering as the primary driver for their organization's focus on the discipline. Of course, systems engineering was still performed during the last century, but as several interviewees stated, "we didn't call it that." As the recognized desire for systems engineering has increased, it follows that more educational programs would be started and more people would graduate from these programs. All senior participants who were asked why they chose not to pursue a master's degree in systems engineering stated that they felt that it was more appropriate for people earlier in their careers, and that their extensive experiences provided more breadth and depth of systems engineering knowledge than would be attained through graduate study. In addition, many of the senior systems engineers without a master's degree in systems engineering (about 80%) still had at least one master's degree in another field and felt that they did not require a second master's degree.

Timing of Master's Degree Attainment

The timing of when a master's degree is attained during a systems engineers career offers insights into the motivation and relevance of obtaining the master's degree.

Master's degrees were most commonly attained within the first half of a senior system's engineer's career. The distribution of the length of time in career before attainment of master's degrees for senior systems engineers can be seen in Figure 36.



	<5 Yrs	5-9 Yrs	10-14 Yrs	15-19 Yrs	20-24 Yrs	25-29 Yrs	30-34 Yrs	35-39 Yrs		
1st Attained	13.10%	23.81%	20.24%	13.10%	5.95%	3.57%	3.57%	1.19%		
2nd Attained	1.19%	2.38%	2.38%	2.38%	4.76%	2.38%	1.19%	1.19%		
3rd Attained	0.00%	1.19%	0.00%	1.19%	1.19%	0.00%	0.00%	0.00%		

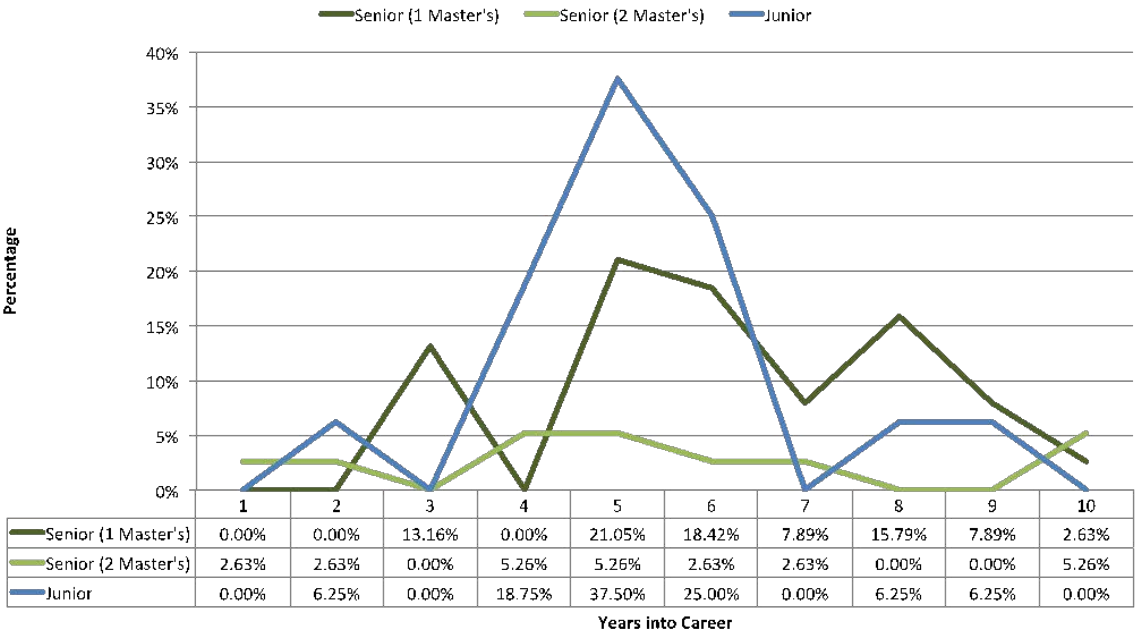
Figure 36. Timing of Master's Degree Attainment for Senior Systems Engineers

The following observations can be made from Figure 36:

- Most senior systems engineers tend to receive a master's degree within the first 15 years of their careers (57%).

- For about 3/4 of those who did attain a second master's, it happened within 15 years of attainment of the first master's degree.
- For those senior systems engineers with a second master's degree, the most common majors were systems engineering (31%) and business or management (31%).
- There are no discernable patterns for the 3% of senior systems engineers with three master's degrees.

A comparison of the first ten years of the careers of senior systems engineers and the careers of today's junior systems engineers in terms of master's degree attainment is illustrated in Figure 37.



	1	2	3	4	5	6	7	8	9	10
Senior (1 Master's)	0.00%	0.00%	13.16%	0.00%	21.05%	18.42%	7.89%	15.79%	7.89%	2.63%
Senior (2 Master's)	2.63%	2.63%	0.00%	5.26%	5.26%	2.63%	2.63%	0.00%	0.00%	5.26%
Junior	0.00%	6.25%	0.00%	18.75%	37.50%	25.00%	0.00%	6.25%	6.25%	0.00%

Figure 37. Comparison of Master's Degree Attainment for Junior and Senior Systems Engineers in The First Ten Years of Their Careers

Almost 2/3 of junior systems engineers who have pursued a graduate degree completed their degree within the first 5 years of their careers (65%); a little more than a 1/3 completed their master's degree 5 years or more into their careers, although none more than 8 years into their careers (36%). Of the systems engineers in the sample who have not completed a master's degree program, two individuals were pursuing a degree when interviewed; almost all of the others indicated during their interviews that they plan to pursue a master's degree in the future.

As discussed earlier, there is a steady growth trend in systems engineering graduate education in the US (Lasfer and Pyster 2013). As the availability of systems engineering graduate education increases, it is reasonable to expect that an increasing number of junior systems engineers would seek a master's in the field. This cannot, however, account for over half of all master's education among junior systems engineers being in systems engineering. Junior systems engineers explained that they sought graduate degrees in systems engineering for several reasons:

- **To Learn Other Ways of Doing Things.** Junior systems engineers often have worked in only one or two organizations. Where they see limitations in the way systems engineering is done in these organizations, they often feel powerless to make changes. By studying systems engineering academically, however, they believe they can better understand alternatives and have a better chance of making an impact on their organization.

- **Broadening of Knowledge.** Junior systems engineers almost unanimously expressed a desire to see different parts of the lifecycle, experience different technologies, understand new techniques in the field, etc. However, these experiences are not always possible in their current organizations or may not be available as soon as the junior systems engineers desire. By obtaining a master's degree in systems engineering, junior systems engineers can at least gain more knowledge on the aspects of systems engineering they have not yet experienced, understand lifecycles from a more holistic perspective, understand multiple processes, and gain exposure to different types of tools. Junior systems engineers who have graduate degrees in systems engineering also expressed concern that their knowledge and skills will atrophy if they are not able to practice them.
- **General Career Growth.** Junior systems engineers generally expressed a desire for growth in their careers, and believed that earning a master's degree in systems engineering would help with this in several ways:
  - Simply obtaining graduate level education in some cases made them eligible for promotions or other incentives that they otherwise would not have had.
  - Systems engineers are often in the position of having to influence engineering decisions, without having authority over the engineers doing the work or making the decisions. Several junior systems engineers stated that having a master's degree gave them at least some level of "street credit".

Analysis of Training

Training, a primary way that organizations attempt to improve the skills of their workforce *en masse,* was consistently discussed by interviewees. Interviewees did not provide a specific definition of training, but the following definition encompasses the ways training was described: organized activity aimed at imparting information and/or instructions to improve the recipient's performance or to help him or her attain a required level of knowledge or skill (Business Dictionary 2016). The discussion of training below reflects uses of the term training, which fit this perspective.

There were three additional ways in which the term "training" was used: as an undifferentiated way of describing education; as a way of describing the overarching capabilities of an individual; or in the term "on the job training". For example, several individuals discussed "training" when referring to degree programs at a university. According to the Helix definitions, this falls under "education", and therefore these instances were coded as part of education instead of training. Examples of the second alternative use of the term include phrases such as, "a mechanical engineer by training" or "I do systems engineering work but I am not trained as a systems engineer." Generally, these same individuals would then describe the organizationally-sponsored training they had received in systems engineering; therefore, in these instances the term "training" or "trained" referred not to specific instances of training but really the overarching career path of the individual. These instances were coded as referring to the career paths of the individuals.

"On the job" training was commonly cited, but nearly 100% of the time this was a euphemism for learning through experience and *not* an actual training program or approach. In these instances, these excerpts were coded for experience. Occasionally this referred to an apprenticeship-type arrangement where the individual was partnered with a more senior person who was more intentionally guiding learning and development. In these instances, this was a blend of training, experiences, and mentoring and was coded accordingly.

Three main aspects of training emerged from the data: types of training; subject matter; and common issues or best practices. Each is described in a subsection below. The analysis is based on 647 excerpts around training from across the complete dataset.

Types of Training

The most common characteristics for the different types of training were:

- Delivery mode – this describes the mechanisms used to provide information to the individual(s) who participate in training. Most common delivery modes discussed were in-person and online modules. There were some self-study modules that were paper based.
- Length – the chronological time allocated to training activities for a particular training activity. The most common times were either less than a day; 1 day; 2 days; 5 days; or 2 weeks.
- Organizational push versus personal pull. This is whether an organization mandates or strongly encourages the training or whether the training is specifically sought by an individual.

These things in combination determine the overall character of the training and there were some combinations of training that were more common than others. The subject matter of the course (see section 4.1.2) sometimes dictates the delivery mode and length, but just as often organizational constraints will impact these factors. Likewise, the Organizational Characteristics will influence whether the training is pushed out to individuals or whether individuals must actively seek opportunities to attend training.

Some common types of training discussed in multiple organizations include:

- **In-person short courses.** These were considered “short” courses because they were chronologically short efforts, unlike semester-long courses. These types of courses occurred in 100% of organizations. Generally, these were 1-, 2-, 5-, or 10-day courses, with 1- and 2-day formats being the most common.
    - 1- or 2- day short courses tended to focus on more discrete subject matter. For example, a particular approach to architecture, an introduction to a specific tool, or teaching points on a particular skill like communication.
    - 5-day or longer courses tend to be for broader training. For example, several organizations had an “introduction to systems engineering” course that lasted a week or more. In-depth leadership training also tended to be involved in the longer format courses.
- **Online training.** Generally this type of training focused on very discrete skillsets and was limited to a few hours. For example, a tool tutorial or an overview of organizational process would be common subject matter for this format. These types of training were usually developed internally. Most organizations had some online component to their training, though all participants who discussed the issue consistently stated that they found organization-sponsored in-person training more effective than organization-sponsored online training.
- **Seminars.** “Lunch-and-learns” or “brown bag” training courses were another common approach. These are informal short sessions, usually no more than two hours on very specific topics.

Typically, a subject matter expert (SME) on the area within the organization will generate and provide the training. Participants indicated that these are informative and useful, but an individual session does not create a major change in proficiency. This type of training was reported in 40% of the organizations.

The language to create the categories above comes from the data, but aligns with delivery modes described in the literature (e.g. Buch and Bartley 2002) with the exception of the seminars. These types of seminars were more commonly grouped with in-person (or classroom based) instruction, but because the subjects of the Helix data view the purpose and outcomes of these types of training differently, they are separated here.

There was a broad mix of "mandated" versus "available" training, though in 90% of organizations there were occasional to consistent refrains that training was not as available as it should be. Even mandated training was sometimes difficult to obtain due to organizational or project constraints. For example, if an individual was deemed "too valuable" on a project, she may be told that the project could not allow her time off to attend even mandatory training, or a mid-level systems engineer might be told by his manager that "mandatory" training was more critical for more junior systems engineers and, therefore, he would not be able to attend. Participants consistently stated that the training they needed was not fully available and that they had consistent support to attend any training they needed in only 10% of the organizations.

Training Subjects

Training can cover a wide variety of subjects and not all of them are related to systems engineering. For this analysis, the Helix team coded only training that was related in some way to the *Atlas* proficiencies. For clarity, the common training subjects reported below are grouped into the proficiency area to which they most closely relate.

- **Systems Domain and Operational Context.** There were some training subjects that addressed elements of the domain(s) relevant to the system and the context in which the system will be used. In general, there was less coverage/conversation in this area than in others.
    - **Engineering Disciplines.** A few organizations provided introductory courses for systems engineers regarding other engineering disciplines, helping to identify "what systems engineers need to know" about software engineering, for example. These were generally one- or two-day courses focused primarily on orienting systems engineers to the terminology, key issues or concerns, and perhaps processes that are more common in the discipline. For example, in a software orientation course, agile methods – which are in many ways antithetical to the way large-scale defense systems are created – might be discussed or a comparison of object-oriented software architecture and functional architecture might be made. Again, this is generally enough to help systems engineers begin to "speak the language" in a less familiar discipline. In Bloom's Taxonomy terms, this would be to help individuals get to the "Knowledge" level, with perhaps "Comprehension" in some basic areas.
    - **Relevant Technologies.** A few organizations provided introductory courses for systems engineers regarding specific technologies that are critical to the systems at hand. These were generally one- or two-day courses and similar to those on Engineering Disciplines, focused primarily on orienting systems engineers to the terminology and key issues or concerns for the technology of interest. For example, in an organization that focused primarily on communications, there were short courses on radar technologies, which

    included the basic terminology, the basic physics of how radar technology works, and critical considerations for radar technologies in development. Again, in Bloom's Taxonomy terms, these courses would be to help individuals get to the "Knowledge" level, with perhaps "Comprehension" in some basic areas and are generally targeted to people who are experienced in other domains, but new to the system of interest.

- **Systems Engineering Discipline.** Perhaps unsurprisingly, one of the most common types of training discussed was training on systems engineering itself, which was offered in some form by each of the organizations participating in Helix, though junior and mid-level systems engineers were more likely to have attended formal training than senior systems engineers.
    - **Foundations.** There were several organizations (~40%) that provided a foundational or introductory "course" in systems engineering. This was often a multi-day course (sometimes up to a 5-day course) that covered the basics of the discipline, provided an overview of the systems lifecycle, introduced terminology, the value of systems engineering, etc. These courses tended to be targeted to new systems engineers transitioning from other disciplines (such as classic engineering) or younger systems engineers who had relatively little experience. Typically, mid-level and senior systems engineers stated that this type of course would not be useful for them because a one-week training course could not match what they had learned from their wealth of experience; however, they often felt it was useful to individuals new to the discipline.
        - Most often, these courses are developed internally and also include "branding" of the processes and methods for systems engineering within the organization, as opposed to being based on a wider community view such as the INCOSE *Systems Engineering Handbook* (INCOSE 2015).
        - In some organizations (15%), there is a shorter version of this course (for example a half-day or one-day course) that is provided to executive leadership, management, project managers, etc. In organizations where this approach was used, interviewees tended to report a more supportive culture or at least reported that the value and acceptance of SE in the organization was improving.
    - **Specific Topics.** In some organizations, only the overview SE course was offered. However, other organizations offered a variety of training courses in SE-related topics. Some common examples included architecture, risk management, decision management, etc. These tended to be shorter courses than the overview courses (often 1 or 2 days) and might include general terminology and knowledge, but would also include the specific approaches used by the organization and perhaps an introduction to any critical tools used by the organization.
    - **Tool-Specific Training.** There were a number of training courses offered around specific tools that support systems engineering work. Again, these tended to be short courses that were very focused on the *use* of the tools and not on the rationale behind the tools. For example, DOORS (IBM 2016) was the most commonly-cited tool and the most-commonly mentioned tool-specific training. But training was described as "how to use DOORS" and not on how to elicit good requirements. Other common tools were architecture-focused tools, MindMapper (MindMapper 2016), and tools to support model-based systems engineering. Overall, this was generally viewed as useful training and in some organizations where the tools were available but *not* training courses, this was listed as a

    critical gap. However, individuals stated that if they did not immediately use the tools on the job, the training became less effective, as those skills would be quickly lost.

  - **Systems Engineering Processes**. Sometimes incorporated into a systems engineering overview course, this was also occasionally offered as a stand-alone course. This type of training would specifically help individuals understand the processes used to conduct systems engineering *in their organization* and may or may not have ties to systems engineering processes such as DoD 5000.02 (DoD 2015). A common shortcoming cited with this type of training was that it trains the process steps but does not explain why the process looks the way they do and does not teach how to tailor the process; both shortcomings were cited in several organizations as areas in which this training could be improved.

- **Interpersonal Skills.** Coverage of training for these types of skills was mixed. Where it was available, this type of training was generally seen as high value in helping develop critical skills for systems engineers that might be underdeveloped in other engineers.
  - **Communication.** Several organizations provide a variety of training designed to improve communications skills. This could cover interpersonal communication, writing, or presentation. Overall, systems engineers tended to report training on interpersonal communication – particularly on how to communicate effectively with diverse groups of people – to be very valuable. Training on writing was cited as most useful when it was on technical writing that specifically supported the systems engineers' work. Presentation training was rare, but valued when available.
  - **Teamwork.** A few organizations reported that training was available that really focused on interpersonal relationships and how to work in a team. Among other topics were creative abrasion and bringing a diverse group to solution. Again, where offered, these types of training were seen as valuable.

- **Technical Leadership.** In the area of technical leadership, as with Interpersonal Skills, training was not available in all organizations but was generally valued when it was available.
  - **Leadership training.** Individuals in 50% of the organizations in the sample reported some form of leadership training. In 15%, this was associated with "high potential" programs – these are often rotational programs that are designed to grow systems engineers – and so are offered to a limited number of people. But in most, there are multiple leadership training opportunities available, though again the approaches vary between organizations. Some have basic modules that can be studied online to deal with specific topics like conflict resolution while others have broad courses available. In a few organizations (10%), there is tiered leadership training that goes from a one-day course for all systems engineers to a one-week course for more senior individuals to a two-week course for the most senior leadership. Individuals who had participated I this training typically reported that is was valuable.
  - **Team building.** As opposed to working in a team, only 5% of organizations reported providing training that actually focused on how to bring a team together and lead a team, versus simply being a productive member of a team.
  - **Related Disciplines.** Technical leadership overlaps with relevant skills from other disciplines, particularly project management and business. In some organizations, systems engineers are encouraged or allowed to take training courses in these areas.

> Project management courses are more readily available, but business courses, which are typically tailored to the business aspects relevant to the context of the organization, were seen as extremely useful.

Though the utility of each type of training varied between organizations, and indeed occasionally between individuals within the same organization, overall individuals were pleased with the opportunity to attend training and learn new things.

In general, the data was not detailed enough to draw specific conclusions about the change in proficiency expected for training in specific subjects. However, among the 97 individuals who performed proficiency self-assessments, training was described as accounting for at most an improvement of 1 or 2 on an ordinal 1 to 10 scale. In other words, if an individual rated herself as having grown in proficiency in an area from "2" to "7", she would not generally attribute more than 1 or 2 ordinal steps in proficiency growth to training. Training can provide some improvements in proficiency, but were in no way considered a substitute for experiences. And importantly, of the individuals who provided insight on best practices, 67% indicated *actually applying learning from training during or soon after a training event is a best practice* and 17% of the issues with training raised indicated that if an individual did not apply the learning from a training program on the job immediately or soon after the training, then the impact of the training could be lost. This is consistent with training literature which shows that training transfer – the ability to apply skills learned in training on the job – is limited when individuals do not have opportunities to use new learning in their work (e.g. Brinkerhoff and Montesino 1995, Lim and Mooris 2006, and Burke and Hutchins 2007). Clarke (2002) states that limited opportunity to perform skills on the job is the primary limiting factor on training transfer.

Training Issues and Best Practices

There were 103 excerpts in which individuals discussed issues, best practices, or risks. Among these excerpts, there were a few issues identified consistently by participants.

- **Training must be "immediately" applied or lost.** Participants indicated training that could be applied on the job during or immediately after training was internalized and remembered much more effectively than training which occurred "in the classroom" only, which was much more likely to need to be repeated. Likewise, in the discussion of proficiency changes over time, individuals indicated that training that was applied resulted in a *long-term* increase in proficiency while simply attending a training course would provide a *temporary* increase in proficiency that would diminish without application. This aligns with findings from the literature about the importance of the "opportunity to perform" (Burke and Hutchins 2007) for training transfer.
- **Access to training.** Across government and commercial organizations and across a variety of domains, a common issue that was cited was access to training. The most common issue was that even when training was mandatory or when it was optional but cited as a priority within the organization, it was often difficult to gain access to training. There were a variety of reasons, but as participants discussed the issues, they came down to a few common causes: a lack of funding for training; a disconnect between organizational and project priority; and lack of managerial support. Particularly in government organizations or commercial organizations in the defense industry still recovering from the 2012-2013 sequestrations, the lack of funds for training was a very common concern. There were many individuals who stated that while the leadership of the organization encouraged training, project managers balked at allowing individuals leave time to attend even partial-day training. Likewise, several individuals stated that while they would

request training, their managers would not support their attendance, though in some instances this was tied to lack of available training funds. A related issue was the dissemination of information about training opportunities within an organization. In other words, individual systems engineers were not always aware of the training courses available, even though the Helix team was informed of these initiatives by the leadership team.

- **Some anticipated risks can be addressed by training.** These included how organizations could become more agile, better incorporate and integrate COTS into design, and MBSE were topics that were mentioned. Common themes were the lack of domain knowledge and the rationale behind the processes used within organization. The former would be a way to help younger systems engineers who would have to mature more rapidly to replace retiring senior systems engineers. For the latter, there were many training opportunities on process, but a common criticism of them was that while the process *steps* are taught, the rationale behind the process – how it came to be, why and when it might be tailored – was not understood consistently, particularly among junior systems engineers.

4.6 PERSONAL CHARACTERISTICS

Personal characteristics can either enable or inhibit a systems engineer's ability to deliver value. They also influence the efficiency of the forces that impact the effectiveness of the systems engineer. However, it is also possible for the characteristics to be influenced by the forces, as illustrated in Figure 38.

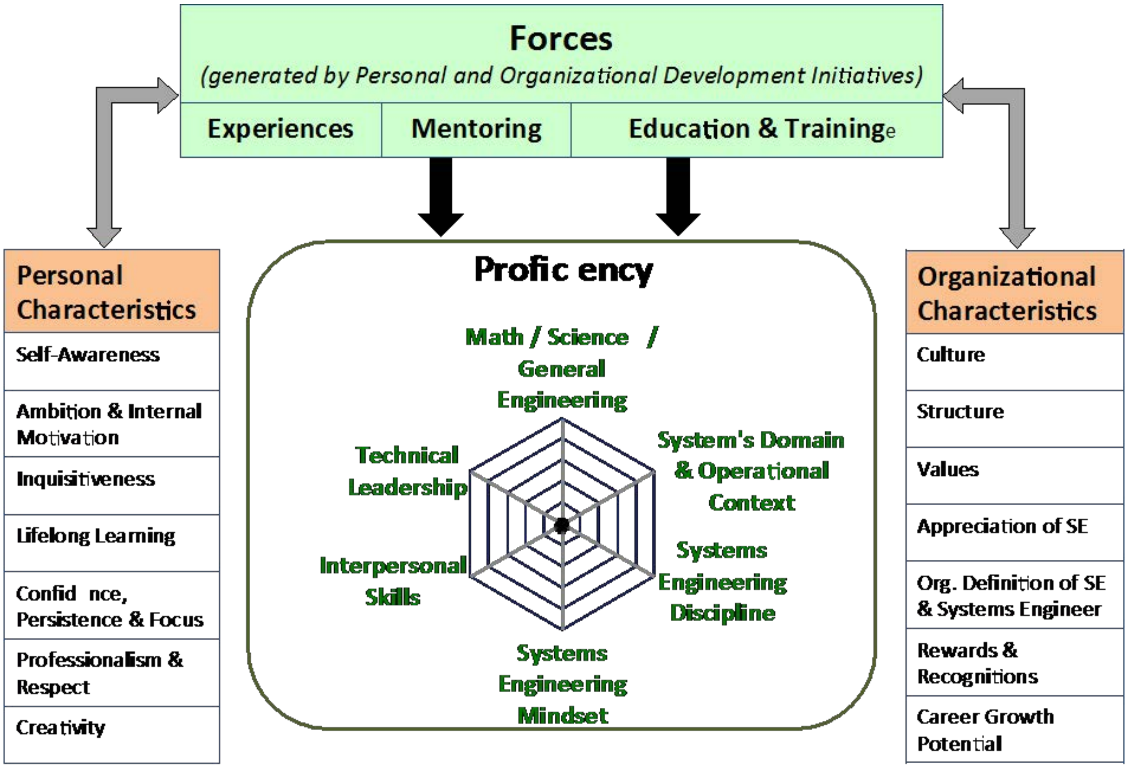


Figure 38. Forces, Proficiency and Characteristics

The proficiency framework provides details on various knowledge, skills, abilities, behaviors, and cognitions that systems engineers need in order to be effective. In addition, there are certain characteristics that systems engineers possess that influence the impact of value delivery; they also influence the impact of the forces on the effectiveness of the systems engineer. For example, two individuals with similar educational backgrounds and experiences undergoing the same training program may accrue different levels of benefits.

These characteristics influence the proficiency of the systems engineer and the impact of the forces on changes in those proficiencies. The characteristics are also different from the proficiencies identified in *Atlas* because the characteristics are focused on the qualities of an individual versus the skills or abilities of that individual. This distinction was sometimes difficult for interviewees to grasp, particularly when they are compared with *Systems Engineering Mindset* categories such as *Big-Picture Thinking* or *Paradoxical Mindset*. The differentiation is that "personal characteristics" are traits of the individual that may *enable* the development of these proficiencies, but they do not in and of themselves ensure that an individual has a certain level of proficiency.

The two proficiency categories that most interact with personal characteristics are *Big-Picture Thinking* and *Paradoxical Mindset.* For example, one pairing in *Paradoxical Mindset* is “courageous and humble”. Two characteristics commonly cited were “courage” and “humility”. These clearly support the category of *Paradoxical Mindset*. However, neither is in itself a guarantee of proficiency. Instead, *Paradoxical Mindset* indicates that an individual both possesses these characteristics and also can make a judgment based on the current situation as to which characteristic should dominate. Finally, while the proficiencies are specific for systems engineers, the characteristics described here are not specific to any field, though they have been cited as critically important for systems engineers. These subtleties will be further explored in future versions of *Atlas*.

The primary personal characteristics are identified below. Some characteristics are combined into larger, related groupings for ease of understanding.

- **Self-Awareness:** Systems engineers need to be able to self-reflect and become aware of their strengths and weaknesses; what they know and don’t know; and where they are right and where they are wrong. Of the feedback Helix received on critical characteristics, 11% focused specifically on the need for self-awareness. An increased awareness of oneself is only the first step; to acknowledge gaps or areas for improvement and to correct them requires humility and modesty as well. Of the individuals who felt self-awareness was critically important for systems engineers, 14% indicated that humility was a critical supporting characteristic. Such self-awareness also brings clarity to a systems engineer’s mind as to where they can rely on their own knowledge and where they need to seek the expertise and experience of others. Another interviewee said, “The best systems engineers I work around are people that don’t think they know everything.” Additionally, 21% of the systems engineers who focused on self-awareness explained that courage was also critical; it takes courage to admit your weaknesses to yourself and others; it takes courage to seek opportunities to improve in these areas. Self-awareness was described as a critically important characteristic for the development of *Big-Picture Thinking* (21%) and *Paradoxical Mindset* (21%).
- **Ambition and Internal Motivation:** Systems engineers tend to be very ambitious in terms of their career goals as well as the nature of the systems they wish to engineer. Many systems engineers who participated in Helix interviews expressed a desire to eventually become a ‘Chief Systems Engineer’ or play another senior role, and to also work on increasingly complex projects. Such high ambitions make systems engineers very strongly motivated internally. Even in challenging situations, they are able to generate energy and motivation within themselves, without relying on an external source, which was highlighted in 8% of the feedback on characteristics. They tend to approach any problem with a ‘can-do’ attitude. A fifth of the individuals who discussed internal motivation stated that it also tends to be correlated with self-awareness; as individuals are motivated to grow and improve, they have to be honest with themselves about their own capabilities in order to determine a best path forward. Ambition and internal motivation were typically described as critical characteristics for the development of *Big Picture Thinking* (62%) and *Paradoxical Mindset* (15%).
- **Inquisitive:** Systems engineers, as a group, tend to be very inquisitive, constantly asking “why” questions and trying to understand how a system, project, or team really works. This is a reflection of the naturally curious nature of many systems engineers, and 10% of the excerpts on characteristics focused around this curious and inquisitive aspect. As one interviewee stated, “I think most engineers who really like engineering are just curious and they want to understand how things work and how things go together and then be challenged to work on and solve different problems. About a third (31%) of individuals who discussed the importance for systems engineers to be inquisitive explained that courage was an important supporting characteristic;

systems engineers who are too timid may not ask the necessary questions and many indicated that inquisitiveness is critical to the development of proficiency in *Big-Picture Thinking* (40%).

- **Lifelong Learning:** Systems engineers need to stay current with recent developments in technologies, tools, and processes. To maintain technical relevance, they must constantly improve their understanding of the system's domain and related disciplines. This means that systems engineers need to be students all their lives, constantly learning and increasing their knowledge, irrespective of their seniority or position in the organization. This was cited as a critical characteristic by 6% of the sample. Related to the ability to learn is also the ability to teach and share knowledge with others. It is common for systems engineers to play the role of a teacher or mentor to members of their team. Lifelong learning was commonly correlated with internal motivation (25%); the desire for growth and change is addressed by continual learning. A quarter of the excerpts on lifelong learning indicated that it was a critical characteristic for the development of *Big-Picture Thinking*.

- **Confidence, Persistence and Focus:** Systems engineers must be confident in their own abilities in order to interact effectively with organizational top management, with subject matter experts who are technically sound, with strong personalities who are highly opinionated, and with end customers whose requirements are to be satisfactorily met for the system to be successful. Systems engineers are not known to be shy, and they do not 'give-up' easily. They remain persistent to make things progress towards the end goal and vision for system success, which is always in their focus; and they do not get distracted easily. (6%) It requires some courage to be confident in one's abilities when dealing with multiple stakeholders and carry one's belief in the correct approach or decisions forward (11%). These characteristics were seen as critically important for the development of *Big-Picture Thinking* (33%), *Paradoxical Mindset* (7%), and effective *Communication (*33%).

- **Professionalism and Respect:** Systems engineers deal with a wide variety of people across the technical, programmatic, and business aspects of their work and often must lead through influence rather than direct authority. It is critical, therefore, that they are professional in their conduct, mannerisms, and behaviors and that they maintain a good work ethic. They treat others with respect, and acknowledge their strengths, contributions, and the value they bring to the team and the organization. They are patient when they need to be, and tend to 'tolerate' difficult people well. In order to achieve this, they must possess a certain level of humility (58%), focus on and respect for the variety of social backgrounds and personalities on their teams (25%), sense of their own accountability to the team (8%), and maintain ethical behaviors (8%). Interviewees viewed these attributes as critical to developing several proficiencies: *Building a Social Network* (50%), *Paradoxical Mindset* (13%), *Big-Picture Thinking* (9%), *Communication* (9%), and *Flexible Comfort Zone* (9%).

- **Creativity**: "Systems engineering is a unique combination [of] left brain – right brain working together," said one interviewee; others also indicated that there was an artistic side to systems engineering. In terms of how creativity could be fostered, most interviewees elaborated not on their professional experiences but focused more on their hobbies, interests, and life in general outside the office, and the influence of all those on their professional life as a systems engineer. In many cases, participants talked about developing the proficiencies in the *Systems Engineering Mindset* category very early on in life, and through personal interests. Among the systems engineers who participated in Helix interviews were musicians, avid readers, poets, painters, authors, puzzle-lovers, dancers, and gardeners. A quarter of individuals who discussed creativity explained that they loved puzzles from an early age and internalized a critical lesson from their

> work on puzzles: that there *is* a solution to any challenge – it may not be obvious, but it exists somewhere to be discovered. Individuals who discussed creativity indicated that it was critically important to the development of proficiency in *Big-Picture Thinking* (50%) and *Paradoxical Mindset* (50%).

The personal enabling characteristics discussed above play a critical role in the development of effective systems engineers. Some interviewees who had been mentors or supervisors of systems engineers stated that these characteristics often serve as indicators for identifying potentially effective systems engineers. Experiences and mentoring appear to have the strongest influence on the development of these characteristics. None of the systems engineers who provided detail on characteristics discussed how education or training supported them.

As noted above, interviewees commonly cited two proficiencies as being well supported by these characteristics: *Big-Picture Thinking* and *Paradoxical Mindset*. Other proficiencies were discussed, but with 24% of the total excerpts on characteristics being correlated to *Big-Picture Thinking* and 11% correlated to *Paradoxical Mindset*, these were the most commonly linked to characteristics. The characteristics that most strongly supported *Big-Picture Thinking* were inquisitiveness (25%) and internal motivation (13%), though 14 other characteristics also were described as supporting development of proficiency in this area. For *Paradoxical Mindset*, self awareness had the strongest correlation (20%) and creativity, curiosity, and humility were each correlated with development of proficiency (13% for each respectively); an additional 6 characteristics were also described as supporting development in this area.

4.7 ORGANIZATIONAL CHARACTERISTICS

Organizational characteristics can either enable or inhibit a systems engineer's ability to deliver value. They also influence the efficiency of the forces that impact the effectiveness of the systems engineer. However, it is also possible for the characteristics to be influenced by the forces, as illustrated in Figure 38 (above).

There are several characteristics of organizations that influence how difficult or easy it may be for a systems engineer to be effective. The primary characteristics are discussed below.

- **Culture, Structures, and Values:** While an organization's overarching culture, structure, and values have a much bigger impact than just on the systems engineering community, these factors certainly impact the ability of systems engineers to provide value to the organization.
    - A culture that values individual contributions over team contributions, for example, is a difficult environment for a systems engineer whose value is often realized through team coordination and interaction.
    - The way systems engineers are placed within the overall organization and how they are deployed to projects can impact performance.
    - Organizations that do state a value proposition for systems engineers tend to make systems engineering training more available and facilitate outreach with other disciplines.
- **Appreciation of Systems Engineering:** If an organization has no value proposition for systems engineers or if the value proposition for systems engineers is unclear, it raises uncertainties with individuals outside of the systems engineering community. These individuals do not understand what their expectations from systems engineers should be or what return on investment they should expect when they allocate a portion of their budget to systems engineering activities.
- **Organizational Definition of "Systems Engineering" and "Systems Engineer":** When an organization has an ambiguous definition of these terms – or no definition – it is an impediment to a systems engineer's effectiveness. In organizations with unclear or no definitions of these terms, individuals outside of the systems engineering community form their own definition of what a systems engineer does based on their personal experiences with an often limited sample of systems engineers. When the title "systems engineer" is applied loosely within an organization, it can cause tension, as people do not have clear expectations of what value a systems engineer should truly bring to a project.
- **Rewards and Recognition:** Organizations tend to have a very common and generic annual performance evaluation system; there are no specific outcomes or objectives related to the value that systems engineers provide. Organizations need a consistent means of evaluating or rewarding systems engineering practice.
- **Career Growth Potential:** In organizations where the career path for a systems engineer is obscure, the discipline is seen as less appealing than other areas where career growth and opportunity is more clearly defined.

4.8 PERSONAL DEVELOPMENT INITIATIVES

When asked what personal initiatives they had for improving their own effectiveness, 100% of the systems engineers in the sample participated in organizational initiatives in some ways – most specifically in mandatory training or mentoring programs. Many fewer individuals had personal growth initiatives (7%) outside of the initiatives of their organizations. There were a few common approaches:

- *Individual Reading* – Some individuals reported that they spent personal time on reading that was related to their work. This included a variety of things, such as journal articles, conference papers, trade publications, relevant news or magazine articles, or books. Journal articles, conference papers, trade publications, and new articles tended to be around technical subjects – new technologies related to the systems the individual supported, classic engineering disciplines, relevant domains, or systems engineering itself (such as the *Systems Engineering* journal or the IEEE *Systems* journal). In terms of books, however, these were more commonly around non-technical areas such as technical leadership, particularly business, or interpersonal skills, particularly communication.

- *Attending conferences* – several individuals stated that they attended conferences relevant to their work whenever possible. This could be a mix of domain-specific, classic engineering, systems engineering, or project management related. Individuals who attended conferences stated that their organizations sponsored their attendance but that this was not a broad initiative; rather, their individual managers or programs helped them find funding to attend relevant events. An additional few individuals indicated that they used to attend conferences, but that funding was no longer available for these efforts and had not been for the last five years or more.

- *Online courses* – these are not full academic courses for credit that could be counted towards a degree. Those types of courses were considered education. However, a few individuals indicated that there were free courses available online, for example massive open online courses (MOOCs) or small, university-sponsored free courses on relevant topics. There are a variety of topics, including overviews of basic classic engineering disciplines such as electrical or software engineering, topics such as risk- or decision-management, or specific technology areas. Individuals who took these courses indicated that they were helpful for gaining an overview, particularly in areas that were relevant to the systems and individual worked on but in which he did not have experience. These courses are free, so participation in them is wholly dependent on the motivation of the individual.

- *Certification* – *All* of the DoD organizations required an engineering certification (at the time of the Helix interviews, the Systems Planning, Research, Development, and Engineering (SPRDE) certification) for all of their systems engineers. However, a few individuals have sought individual certification. None of the organizations specifically sponsored external certification initiatives and the few individuals who had become certified said that they did not believe that it would help them in their organizations but they felt it was important for them as individuals. The three types of certifications discussed were INCOSE Certified Systems Engineering Profession (CSEP); PMI Project Management Profession (PMP); and state-certified professional engineer, which was not specific to systems engineering.

Of the individuals who stated they did not do anything outside of organizational initiatives, many junior and mid-level systems engineers said that they would like to, but that there are roadblocks. The most common are that their work responsibilities are time-consuming enough that they do not have time for extra work or that their managers are not supportive of individuals pursuing additional training. In at least one organization, individuals stated that they were expected to pursue training but were not given leave from their roles and were “dinged on their performance” for this. Most of the senior systems engineers who discussed personal initiatives stated that beyond reading or attending conferences, they did not believe they needed to do more than continue to build upon their experiences. However, almost 5% of senior systems engineers had at one point taken the initiative to create training programs to pass on their knowledge and experiences to younger systems engineers in their organizations.

4.9 ORGANIZATIONAL DEVELOPMENT INITIATIVES

Helix identifies 'initiatives' (both personal and organizational), as those that are intended to generate one or more the forces (experiences, mentoring, and education & training) in a direct manner. These forces, in turn, are expected to improve the proficiency of an individual systems engineer. This section presents various aspects of organizational development initiatives that were discussed during Helix interviews, with a particular focus on initiatives that are available for the benefit of the systems engineers in the organization.

The discussion presented in this section is aggregated from the 40% of all Helix interviews in which participants discussed organizational initiatives. In organizations with a larger number of Helix participants, a richer view of the organization emerged, sometimes with conflicting views presented by the participants. While these are highlighted in the discussion, the intent is not to provide an organization level analysis of initiatives.

4.9.1 NATURE OF ORGANIZATIONAL INITIATIVES

Many features of organizational characteristics can be observed from Helix interviews:

- **Distinction between initiatives and policies**: It is not always straightforward to recognize and identify organizational initiatives, and to distinguish them from organizational practices and policies. Helix considers it an initiative if the organization plays an active role in promoting, enabling, and supporting it for the benefit of its employees. For example:
    - Some organizations provide tuition reimbursement to their employees seeking graduate degrees in related disciplines, subject to policies regarding eligibility, absence from work, etc. Typically, it is up to the individual employee and her immediate supervisor to take advantage of those policies.
    - Other organizations play a more active role in providing graduate education for their employees: they establish relations with specific universities; they establish cohorts for individual courses and/or degree programs; they provide facilities within their premises for the universities to conduct courses; they make available organizational data for projects and dissertations; and also tend to reward employees who go through these programs with a promotion or salary raise.
- **Scope of organizational initiatives**: Some organizational initiatives are targeted at systems engineers' proficiencies, systems engineering proficiencies of the workforce, or within the systems engineering department/division. There are initiatives that are offered only to those systems engineers that meet certain eligibility criteria and not to the entire systems engineering population. These "high potential" programs are generally intended to help selected systems engineers mature more rapidly. There are also other initiatives intended for the benefit of all employees across the entire organization, which include any systems engineers; for example, some organizations will pay for any graduate education, regardless of subject. Each of these can be a benefit to a systems engineer, though programs scoped specifically to the systems engineering population tend to be more directly beneficial.
- **Influence of organizational initiatives on organizational characteristics:** While some organizational initiatives generate forces that in turn improve the proficiency levels of individual systems engineers, some other organizational initiatives improve organizational characteristics – either directly or indirectly. For example:

  - Some organizations have initiatives to identify and recruit SE talent from within the organization, and also to recognize and reward achievements of systems engineers and other employees. Such initiatives do not directly improve any of the forces, but rather the organizational characteristics.
  - Some organizations have mentoring initiatives to develop their junior systems engineers by pairing them up with senior systems engineers. Such initiatives are intended to directly benefit the mentee. However, such relationships between junior and senior systems engineers also tend to improve the environment and culture of the organization. (See Section 6.2.4 on the benefits of mentoring.)
- **Formal and informal initiatives**: By definition, organizational initiatives are formally established and deployed. However, there are also informal versions of those formal initiatives that could even co-exist with formal versions within the same organization. Some informal initiatives are also established by the organization. For example:
  - It is typical for mentors and mentees to form an informal mentoring relationship, without being explicitly directed by the organization. Such informal mentoring relationships tend to exist irrespective of the establishment of a formal mentoring initiative in that organization.
  - Some organizations offer a variety of training courses on topics of relevance, often in a classroom setting. In addition, there are also informal training and information sessions that the organization offers – as guest lectures or lunch-and-learn programs.
- **Portfolio of initiatives**: Organizational initiatives rarely exist in isolation; typically, a portfolio of initiatives is available to employees. Organizations establish individual initiatives to address various needs; and in some cases, a higher-level initiative leads to many lower level initiatives as well. For example, an organization may have mentoring and rotational programs. These may be linked, such that each new rotation pairs an individual with a new/additional mentor. An individual in the rotation program, then, not only gains skills from new work experiences, but also develops a larger network of trusted individuals on whom she can call for advice and support.

  As another example, an organization may have a goal to increase the percentage of the workforce with graduate degrees and creates an incentive program for graduate education, paying for tuition and giving an individual a number of paid hours each week to devote to study. If many systems engineers take advantage of this to gain formal systems engineering education and the organization identifies clear positive impacts, the organization may decide to partner with a university to develop a cohort program for systems engineering master's education.

4.9.2 TYPES OF ORGANIZATIONAL INITIATIVES

Participants in Helix interviews discussed the features, benefits, and shortcomings of many organizational initiatives that they had either directly participated in or have been aware of – both in their current organizations and in their previous organizations. The many initiatives mentioned, may be classified under the following types:

- **Recruitment initiatives**: These initiatives recognize systems engineering talent and bring individuals into the systems engineering fold. In some organizations, such initiatives bring in new employees from outside the organization – usually fresh graduates or others with limited experience. Other organizations have initiatives to recognize and recruit systems engineers from elsewhere in the organization, usually after a manager has identified the person as a "systems thinker”.

- **Orientation initiatives**: Some initiatives are exclusively targeted at new employees to familiarize them with the organization, its processes, and the way it does systems engineering. In most organizations, a job rotation program is usually offered only to new / junior employees, offering them a glimpse into various parts of the organization before assigning them to one part of the organization. Some organizations recognize the value of such initiatives to senior employees, and extend those initiatives to them as well.
- **Experience enhancing initiatives:** Junior systems engineers grow into senior experienced systems engineers not just by the number of years they spend in an organization, but through performing in various systems engineering roles; different projects; various levels and types of systems; and different phases of a systems lifecycle. Organizations establish initiatives that are designed to effectively provide rich experiences to systems engineers. Typically, these take the form of rotational programs with specific paths depending on the types of skills to be developed.
- **Mentoring initiatives**: These initiatives are very prevalent in many organizations – either as a formal or an informal arrangement. While the primary beneficiaries of mentoring arrangements are the less experienced mentees, the more experienced mentors and the organization at large stands to benefit as well. From a Helix perspective, ‘mentoring’ is also identified as a force that directly impacts and enhances the proficiency of systems engineers. Section 6.2 provides additional discussion on mentoring and mentoring initiatives.
- **Education and training initiatives:** Every employee enters any organization with some level of formal education. Recognizing the value of formal education, many organizations offer many initiatives for their employees to obtain higher degrees from universities. There is also a need for employees to be trained in particular specialized topics, and organizations typically offer many training options of varying types and durations for the benefit of its employees. Various aspects of training are discussed in Section 6.3.
- **Knowledge management initiatives**: A significant risk in many of the organizations that participated in the Helix interviews was the imminent loss of senior system engineers and their vast experiences. Many organizations have established initiatives to capture those experiences in various ways, and to store them in a readily accessible manner as when required.
- **Leadership development initiatives:** The most senior technical position that a systems engineer can achieve in an organization is that of a chief systems engineer or equivalent. Organizations tend to identify high-potential employees from its pool of junior and mid-level systems engineers, and offer them initiatives to enhance their leadership proficiencies in addition to technical proficiencies, thus enabling those systems engineers to develop in to future chief systems engineers and other senior systems engineering positions.
- **Rewards and recognition initiatives:** As a way to motivate, encourage, and appreciate the achievements of its systems engineers, organizations establish various rewards and recognition initiatives specifically for systems engineers in addition to its employees at large.

Overall, initiatives are focused on helping individuals develop additional proficiency using one or more of the forces identified in *Atlas*. For example, rotational programs are designed to increase the breadth of experiences. Apprentice programs – where an individual is paired with a more senior individual and shadows them – provides an opportunity for building proficiencies through both experiences and mentoring. Rewards initiatives generally help to identify and provide solid examples of effective systems engineers, highlighting the key systems engineering values for the organization.

4.9.3 PHASES OF ORGANIZATIONAL INITIATIVES

Helix interview data indicates that organizational initiatives tend to have various phases. Appropriate recognition and management of initiatives across these different phases is critical for success.

- **Identifying the need:** The first step in any organizational initiative is to clearly articulate the need for one, or define the problem that needs to be solved. While there are many types of initiatives that an organization could potentially establish, it is imperative for an organization to understand why a particular initiative is required.
- **Establishing the initiative**: Once the need is recognized and the type of initiative is identified, the organization must then establish the initiative by setting up the required policies, guidance, personnel to run / manage the initiative, criteria for selecting beneficiaries, and the required infrastructure.
- **Deploying the initiative:** There are a number of activities to be done once the organization has established an initiative:
    - *Promoting:* In 90% of the organizations that participated in Helix interviews, there were initiatives that were wholly unknown to at least one Helix interviewee. The organization must take an effort to let its employees know of any initiative that they can benefit from. Newer employees who go through some sort of an orientation tend to be more aware of initiatives that they can immediately benefit from. Even those employees who have spent many years in the organization are not very aware of the initiatives that are available to them.
    - *Enabling*: When an employee is interested in a particular initiative and is eligible, the organization must enable the employee to benefit from that initiative. Experiences shared by Helix participants indicate that there are situations when they are unable to take advantage of an organizational initiative since they could not take time off their regular work to participate in a training initiative, or that some procedures diminished the effectiveness of the initiative.
- **Responding to outcomes of initiatives**: When an employee participates and benefits from an initiative, typically, there are new skills or knowledge that are acquired, and the employee could recommend improvements based on this. For example, if an employee receives education or training on systems engineering processes, and if the organization does not support modification of existing systems engineering processes, it defeats the purpose of the education.
- **Evaluating the initiative:** The most critical aspect of the success of an initiative is to evaluate it periodically, and to then update, reform, stop, or restart an organizational initiative. A critical evaluation could also reveal enablers and inhibitors for the initiatives. Helix interviews indicated evidence of many situations:
    - Initiatives no longer address the need for which they were established.
    - The need for which an initiative was established is no longer valid.
    - There are more trainers than trainees.
    - Employees are not motivated.
    - The evaluation of some initiatives makes it appear more successful than it really is.
    - The procedures and policies for an initiative could be burdensome.

  - There is a need to restart an initiative that used to be very effective but was stopped due to many reasons, including budget cuts.
  - The duration of a training course may be altered.
  - The target beneficiaries for an initiative need to be redefined.

4.9.4 CRITICAL FACTORS FOR SUCCESS WITH ORGANIZATIONAL INITIATIVES

When individuals discussed successes and failures with organizational initiatives, there were four factors that stood out as critical to the success of any initiative:

- **Establishing the right initiative:** Like in any good systems engineering development, identifying the requirements and addressing them appropriately while establishing the initiative is a necessary first step.
- **Spreading the word:** Any organizational initiatives will be ineffective when an intended beneficiary is unaware that such an initiative exists within the organization. Organizations must take an effort to let its employees know about their eligibility and existence of any organizational initiatives, and enable them to benefit from them.
- **Periodical evaluation of the initiative:** Due to the dynamic nature of the organizational environment, it is important to critically evaluate any initiative periodically to identify modifications that need to be made to the initiative.
- **Commitment from leadership:** Even if many relevant and effective initiatives were available, commitment from the organizational and immediate leadership is essential for an employee to benefit from an initiative.

5 Systems Engineers’ Career Paths

In addition to understanding the overall characterization of the elements of a systems engineer’s career, it is helpful to look at the order and overlap of these elements, which can provide additional insights. Helix developed a way to visualize a career path, which is illustrated in Figure 39 below.

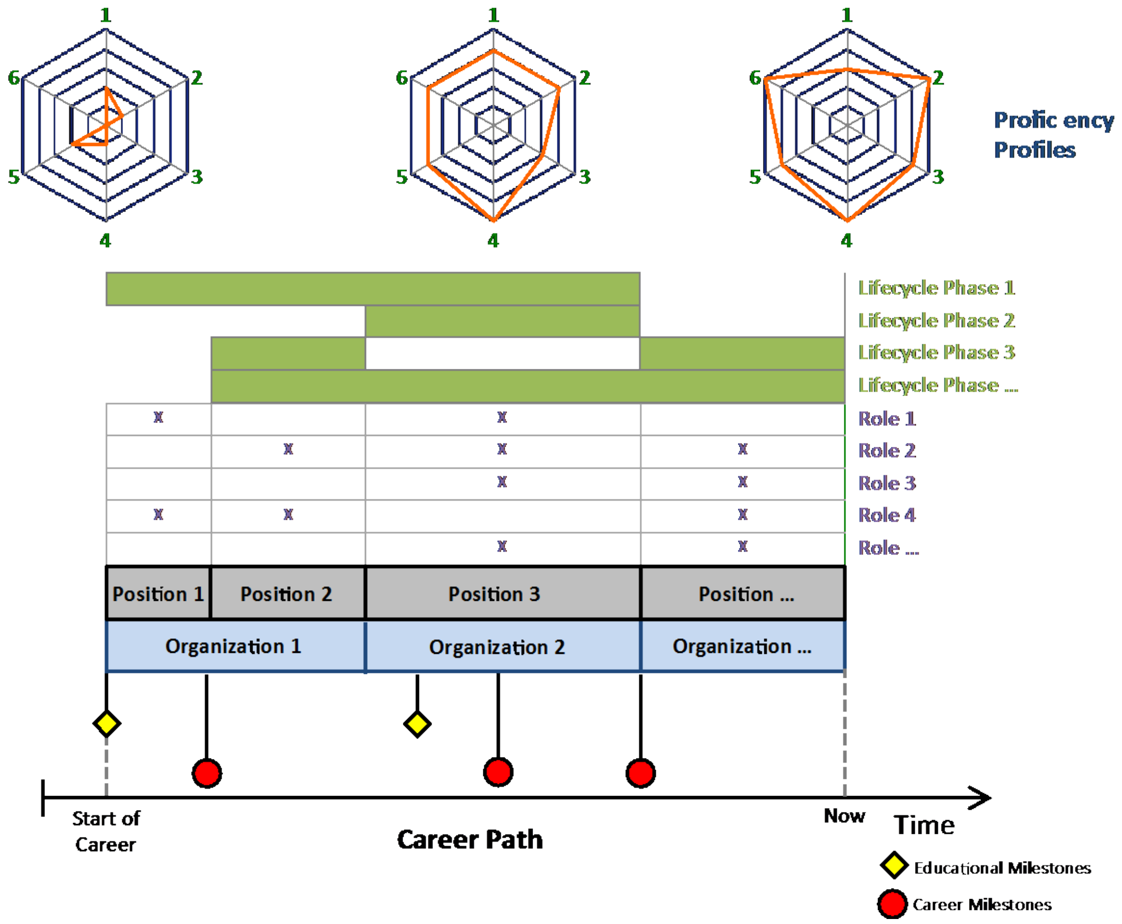


Figure 39. Visualizing a Career Path

The visualization pulls together the following elements of a career path:

- **Timeline:** Time is the dimension onto which all other elements of the career path are projected and visualized. The visualization helps understand the sequence, timing, and duration of various elements of the career path, offering valuable insights for developing the careers of systems engineers.
- **Educational Milestones:** The career of an individual typically begins when an undergraduate degree (or in some cases, a higher degree) is obtained. When, in which disciplines degrees are obtained, and how they impact other elements of the career path can be observed.
- **Career Milestones:** Significant milestones in terms of types of systems engineering positions, such

as first leadership position, chief systems engineer, or program engineer, etc.

- **Organizations:** The variety of organizations and the time spent in each of those organizations can provide interesting insights, particularly if the organizations vary in terms of sectors, key domains, or other factors.
- **Positions:** The number and duration of all positions held across organizations are captured in the career path. How duration and educational qualifications affect positions can be observed.
- **Roles:** The roles performed in each of the above positions perhaps offer the most interesting insights into a career path. An individual is likely to perform more than one role in any particular position, but those roles typically vary as one's career progresses. Some roles performed earlier in one's career may no longer be performed, and there may newer roles that one plays later in her career. The types of roles performed concurrently offer insights into each position.
- **Lifecycle Phases:** The lifecycle phases experienced during each of the above positions are indicated along the career path. The duration and sequence of the lifecycle experiences indicates the exposure that an individual possesses. Similarly, some roles may be more relevant to particular lifecycle phases.
- **Proficiency Profiles:** The level of proficiency can be profiled at any point in the career with respect to the *Atlas* Proficiency areas. Possibly the most difficult to depict accurately across the career, the proficiency profile can be mapped onto the roles and positions that one performs. This visualization also helps to show, and how education and experiences influence proficiency.

The career path visualization currently does not include mentoring or training, but gathering all elements into a single visualization provides a holistic view of the entire career of an individual systems engineer. When multiple career paths of different individuals are visualized, patterns can be observed that can offer interesting insights for career development of future systems engineers.

5.1 EXAMINING THE CAREERS OF CHIEF SYSTEMS ENGINEERS

The description of a career path is only useful to the extent that it provides valuable insights about how individual systems engineers can grow, mature, and develop their own careers. But what are the paths that lead to success? From the Helix interview data, a Chief Systems Engineer (CSE) is one of the most senior technical positions that a systems engineer can achieve. Individuals who became CSEs were able to do so because they had proven themselves to be effective throughout their careers, and had continually demonstrated the ability to take on increasing responsibilities. Hence, Helix considers the careers of CSEs worthy of further examination, since it can provide valuable insights to systems engineers early in their career to be develop into CSEs in future.

Helix identified 22 individuals in the interview sample who currently hold or have held the CSE position, for further analysis. Though many aspects of education and experiences were explored, a select few which provided particularly strong impacts on proficiency are discussed here: overall educational background; experiences across systems engineering lifecycle phases; and experiences across systems engineering roles.

In addition, there was a wealth of information available from the INCOSE SEP applications, especially on individuals applying for Expert Systems Engineering Professional (ESEP) certification. That information from INCOSE SEP data was also analyzed by Helix.

5.1.1 EDUCATIONAL BACKGROUND OF CSEs

Each CSE in the sample had a bachelor’s degree; for 23% of interviewees, this was the highest degree attained. Around two-thirds of CSEs (64%) held at least one master’s degree and 9% held a PhD. The most common majors for bachelor’s degrees and master’s degrees are shown in Table 10.

Table 13. Bachelor’s and Master’s Degree Majors of CSEs

Degree Majors	Bachelor’s	Master’s
Electrical Engineering	45%	21%
Mechanical Engineering	18%	4%
Civil Engineering	5%	0%
Computer Engineering/Science	5%	8%
Aero/Astro Engineering	0%	4%
Systems Engineering	5%	8%
industrial Engineering	9%	0%
Physics	5%	0%
Mathematics	5%	4%
Business	5%	38%
Other	0%	13%

The following observations can be made from Table 10:

- 85% of CSEs have bachelor’s education in engineering fields; a small number were educated in the physical sciences, mathematics, or business (<5% of each).

- Bachelor's education in systems engineering was also seen in less than 5% of CSEs. It was very common in the overall Helix sample for systems engineers to start out in specialty engineering fields, and the educational backgrounds of CSEs indicate that this was true for them as well.
- In general, engineering bachelor's education prepared CSEs with sufficient proficiency in *Math/Science/General Engineering* to perform detailed design work, do detailed analysis, or support test and evaluation.
- Of the nearly 2/3 of CSEs with master's degrees, 43% earned more than one such degree.
- Only 8% of the CSEs have a master's degree in systems engineering; this is considerably lower than the overall rate of systems engineering graduate degrees in the total Helix sample (26%).
- Most CSEs indicated that they believed their experiences were sufficient and they did not believe that they would benefit enough to warrant the effort required to earn a master's degree in systems engineering.
- About a 1/3 of CSE's master's degrees (38%) were in engineering fields outside of systems engineering; this is lower than what is seen among other senior systems engineers in the sample (50%).
- The most common master's field among CSEs was related to business (38%); generally, these were MBA degrees, though occasionally they were master's of science degrees related to more technical fields such as technology management. Less than half of these (44%) were second master's degrees. The CSEs with these degrees explained that they felt they had sufficient technical understanding but needed to learn more about business, management, finance, and other disciplines that support understanding business processes.
- Doctoral degrees were less common among CSEs (9%) than in the other senior systems engineers in the sample (12%). The most common PhD concentration in the overall sample was systems engineering, but there is no single common field of doctoral study among CSEs; electrical engineering, geotechnical engineering, and atmospheric sciences have equal representation. Doctoral studies were not required for advancement for any of the CSEs. Instead, those with PhDs indicated their continued desire to learn and grow and improve their understanding of specific disciplines was their motivation.

5.1.2 EXPERIENCES ACROSS SYSTEMS ENGINEERING LIFECYCLE PHASES FOR CSES

All of the CSEs in the sample have experiences across either four or five lifecycle phases, but none of the CSEs have experienced all six of the lifecycle phases. Figure 40 provides insight into the order in which CSEs experienced the systems lifecycle.

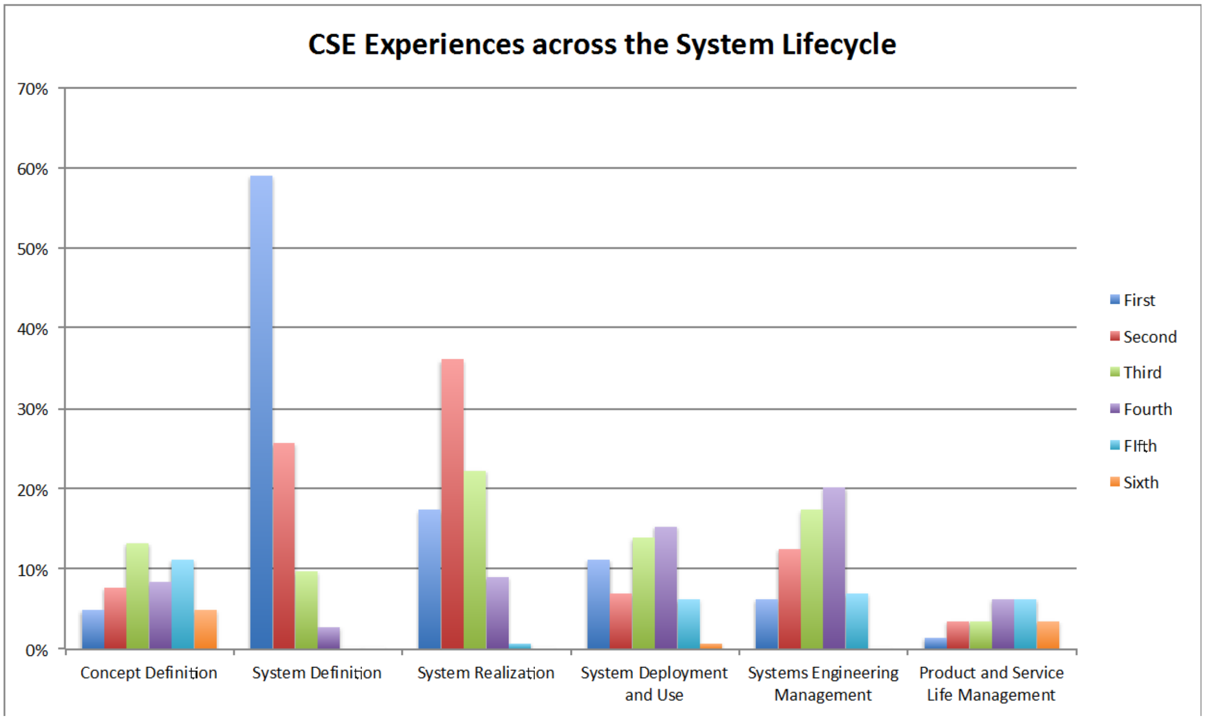


Figure 40. Order of Exposure to Lifecycle Phases Experienced by CSEs

There were a few clear patterns in how CSEs moved through the systems lifecycle process activities:

- All CSEs have experienced *System Definition*, *System Realization*, and *Systems Engineering Management*. This is a higher rate than seen in the overall Helix population (98%, 86%, and 64%), respectively.
- The most common point of entry for CSEs was *System Definition*; it was either the first or second aspect of the lifecycle experienced by 82% of CSEs. The most common pathway for entry into systems related work was through work as a specialty engineer. This detailed work was necessary to gain some depth – to understand how things "really work" and the problems that can be encountered when they try to design something. These *Experiences* impact the *Math/Science/General Engineering* and *System's Domain and Operational Context* proficiency areas. Also, many CSEs experienced leading design work early in their careers, which would improve the *Technical Leadership* and *Interpersonal Skills* proficiency areas.
- Half of the CSEs had experienced *System Realization*. Systems engineers who had experience in manufacturing, which falls into *System Realization*, explained that these *Experiences* are valuable because they help engineers understand the practical considerations and issues of implementing a design. Understanding the basic constraints on the common manufacturing techniques was stated as very valuable in improving design work and limiting the need for redesign.
- For *Systems Engineering Management*, all CSEs in the sample were also *Technical Managers*. The related activities: planning, configuration management, decision management, etc. are all part of the CSE position. In general, these types of activities were reported to help in the development of *Technical Leadership*, *Interpersonal Skills,* and *Systems Engineering Discipline* proficiencies.
- The vast majority of CSEs have experienced *Concept Definition* and *System Deployment and Use*

(85% and 70%, respectively). This is a higher rate than seen in the general Helix population (50% and 55%, respectively).

- *Concept Definition* includes working directly with stakeholders to identify the problem and "true needs" (as opposed to a stakeholder's assumptions about the right type of solution). Gaining this type of understanding first-hand gives systems engineers the opportunity to improve their understanding of the vision for the system and how it will be used, supporting growth in *System's Domain and Operational Concept* and *Systems Engineering Mi*ndset proficiencies. Communicating directly with customers also enables systems engineers to build their skills not just in general communication, but also in the translation of non-technical information for a technical audience and vice versa. This provides an opportunity for improving *Interpersonal Skills* proficiencies. The majority of CSEs have had these *Experiences* – and often fairly early in their careers –helping them grow as systems engineers.
- Considerably more CSEs started in *System Deployment and Use* than did other senior systems engineers (23% and 10%, respectively). Among CSEs, 60% of those who started in *System Deployment and Use* gained experience in the operation and maintenance of relevant systems as members of the US military. The remaining 40% also worked as operators and maintainers, but *working* in industry. Through these *Experiences*, CSEs had the opportunity to understand how a system should operate, what the common processes and procedures were in relation to a system, and to understand the problems that existing systems have. All of these activities provided opportunities to improve the systems engineer's *System's Domain and Operational Concept* proficiency. They also provide key insights about the overall lifecycle of a system, which can improve *Systems Engineering Discipline* proficiency. All of the CSEs who began their careers in *System Deployment and Use* stated that the understanding of issues that can lead to maintenance problems and issues encountered with operating these systems such as counterintuitive interfaces gave them better insights when they eventually began doing design work. These insights better enabled them to do technical tradeoffs and also helped them to better understand the importance of working through a systems concept of operations (CONOPS) early in the design phase. These *Experiences*, then, can improve their proficiencies in *System's Domain and Operational Concept* and *Systems Engineering Mindset* (*Abstraction* and *Foresight),* as well providing specific insights into lifecycle considerations for *Systems Engineering Discipline*.
- No matter where they started in the systems lifecycle, CSEs cited benefits in later phases they experienced. For example, CSEs starting in testing (*System Realization*, 9%) stated they gained insights into the unintended consequences of certain design decisions and these insights helped them avoid some of these pitfalls when they began design work (*System Definition*). Starting in *Concept Definition* – working on stakeholder needs and CONOPS – provided an opportunity to better understand the "end state" or "big picture" – and this helped keep the system goals in mind during the design process.

In the sample of CSEs, there do not seem to be standard patterns to move through the systems engineering lifecycle, except that starting in *System Design* is most common and those who do not start in system design most commonly next move into *System Design*. The order seems less critical than having a mixture of *Experiences* across the lifecycle and having a mindset that enables systems engineers to draw connections across these *Experiences* to enable understanding and growth.

5.1.3 EXPERIENCES ACROSS SYSTEMS ENGINEERING ROLES FOR CSES

There are multiple types of roles that systems engineers can play within a single position or even a single phase of the systems lifecycle. To better understand how individuals grew into their CSE positions, the roles played by CSEs in the Helix sample prior to their first CSE position, and during their first CSE position were analyzed. All roles played by CSEs throughout their careers, up to the point of their participation in Helix interviews, were also analyzed.

Roles Played Prior to First Chief Systems Engineering Position

Figure 41 provides an overview of the roles played by CSEs *prior* to their first CSE position, to provide insight into the career paths that helped these individuals become CSEs. This has been updated to reflect the *Atlas* 1.0 role framework, though the roles of *Systems Engineering Champion* and Concept Creator are not included as the data was not consistently collected across the sample. Note that the roles of *Concept Creator* and *Systems Engineering Champion* are not included in the analysis. This is because data on these roles was not collected consistently throughout the sample; additional follow up will be required to enable analysis for these roles.

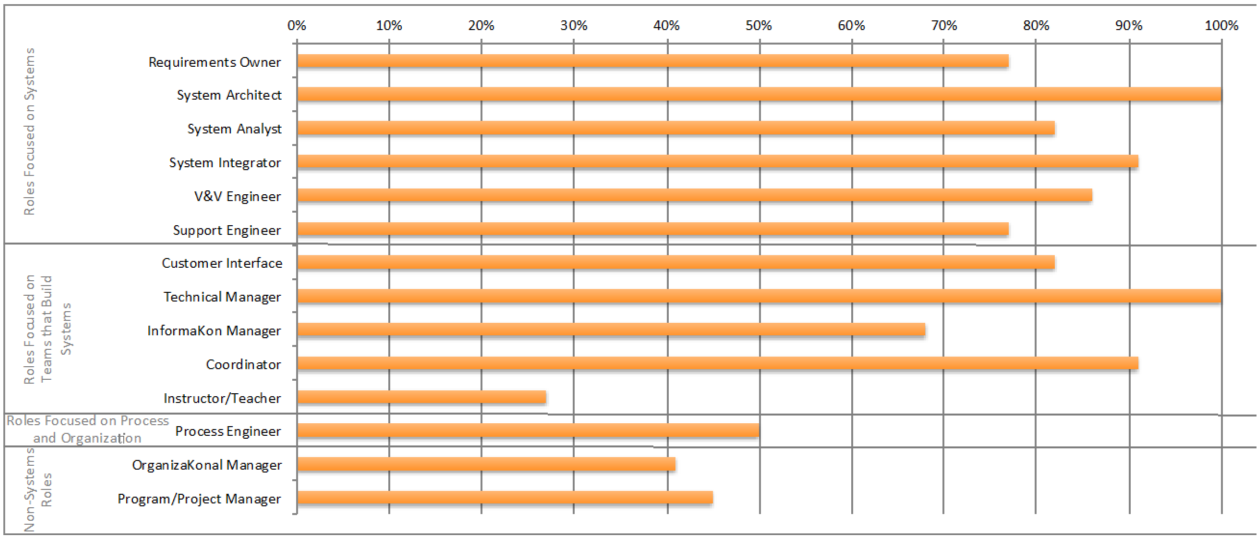


Figure 41. Roles Played by CSEs Prior to Their First CSE Position

Insights from analyzing the roles played by CSEs prior to their first CSE position include:

- All of the systems engineers who would become CSEs worked as *System Designers* and *Technical Managers* prior to their first CSE position. As discussed earlier, these roles are generally a critical aspect of the CSE position, so it is reasonable that individuals would have to prove their abilities in other roles prior to being offered a CSE position.
- The less common roles (50% or lower) are *Process Engineer*, *Organizational Manager*, *Instructor/Teacher*, and *Program/Project Manager*. It is possible that CSEs did work in these areas, but this simply did not make it into their descriptions or discussions of the positions they've played.
- The roles of *Organizational Manager*, *Instructor/Teacher*, and *Program/Project Manager* are generally less common in the Helix sample (26%, 10%, and 15%, respectively). The rates among CSEs are more than twice that seen in the general sample.
- One CSE stated that he had performed the *Organizational Manager* role as a favor to the organization – to fill a role that was needed as an interim measure – but with the expectation that

he would then pursue a technical track. Other CSEs explained that in their organizations, spending some time as an *Organizational Manager* is a requirement before one can become a CSE. Time spent as an *Organizational Manager* primarily provides insights into the functioning of the organization, and may also provide some insight into processes and opportunities to grow a network of experts within an organization, corresponding with growing proficiencies in *Technical Leadership*.

- Over a quarter of the CSEs have been instructors of in-house training or professors at universities focusing on teaching systems engineering or related subjects. These roles improve proficiency in the subject matter not just through the creation of course materials but also through interactions with the students and the application of real-world *Experiences* in an academic setting.
- The role of *Program/Project Manager* has been played by nearly half the CSEs prior to their first CSE position, as in many organizations the *PM* role was considered a role with higher responsibility than that of CSE. In these cases, the CSE acted as a *Program/Project Manager* on a smaller or less complex system before taking on CSE positions on a larger and/or more complex system. CSEs explained that playing this role helped them to better understand not just the technical constraints of a system, but also to build an appreciation for schedule, budgetary, and resource constraints as well as the overall business case for systems development, and also provided the opportunity to understand the customer's perspective in a different way. This role was particularly strong in helping develop proficiency in the *Technical Leadership* area.

The final aspect of the roles of systems engineers is more general than Sheard's twelve roles (1996) and that is the overarching concept of leadership. Each CSE described having several leadership positions early in their careers, starting generally as small group leaders on simple tasks and continually taking on increasing leadership roles throughout their careers. Several of the systems engineering roles described above may have a distinct leadership component. For example, a *Requirements Owner* may start by simply recording requirements in a database, progress to leading a small team to manage the database, then progress to having responsibility for coordinating with the customer (adding the *Customer Interface* role) to generate the best set of requirements while overseeing the team that manages the requirements. These types of patterns – with leadership responsibilities even in more detail-oriented roles – is a common pattern for CSEs, indicating that it is through leadership activities that systems engineers can provide their greatest value.

There were no clear patterns in the order in which CSEs experienced these roles, except that roles such as *Detailed Designer*, *System Analyst*, and *V&V Engineer* were commonly seen earliest in their careers, with roles containing additional scope of responsibility such as *Technical Manager*, *Coordinator*, and *Glue* becoming more common after the first few positions. These roles came earlier in the careers of CSEs than in those of other senior systems engineers, who tended to spend more time in more detail-oriented roles before taking on their first leadership-related positions. Another take-away is that the vast majority of the positions filled by CSEs are multifaceted: they encompass several different systems engineering roles, and this is true from early in their careers.

Among Helix interviewees, it was far more common that multiple roles were played within a single position than that a position included only one type of role. For example, in one organization, a CSE was described as someone who functioned as a *Technical Manager*, *Glue*, *Coordinator* among engineers and mangers, *Customer Interface*, and *Information Manager* at a minimum. The CSEs in the sample held a combined total of 279 positions over their careers; in only 39 of these positions (14%) did the CSE play a single role. Generally, single-role positions were detailed work early in an individual's career, such as working on a requirements database, doing detailed design, or conducting a specific test. When single-role positions occurred later in an individual's career, the roles tended to be *Organizational manager, Instructor/Teacher,* or *Program/Project Manager*.

Role Played during First Chief Systems Engineering Position

The roles described above were those that enabled a systems engineer to grow sufficiently in capability to be offered a first position as CSE. The Helix team then examined the descriptions of the first CSE position provided both in resumes and interview data, and identified the role(s) associated with these positions and their commonality within the sample of CSEs, as shown in Figure 42. Note that the roles of *Concept Creator* and *Systems Engineering Champion* are not included in Figure 40; this is because the data on these roles was not collected consistently and additional follow up will be required.

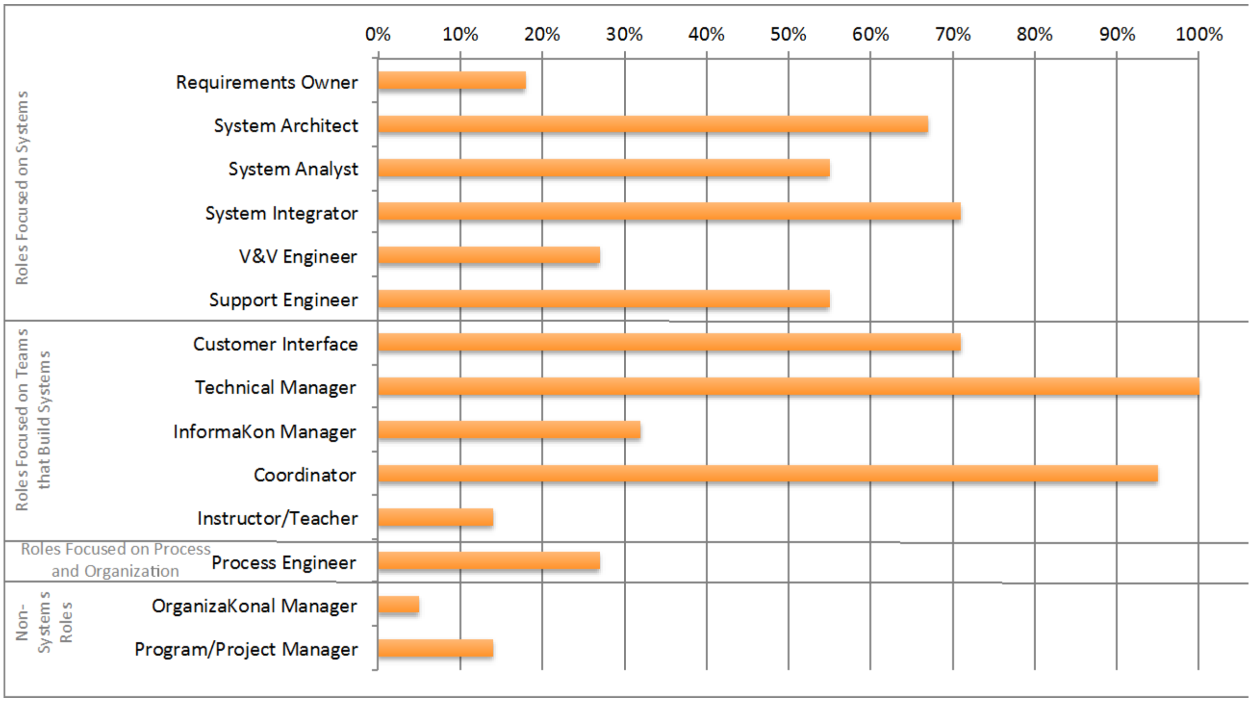


Figure 42. Roles Played by CSEs During Their First CSE Position

Insights from analyzing the roles played by CSEs prior to their first CSE position include:

- All of the CSEs describe the role of *Technical Manager* as a part of the CSE position, and 95% described the role of *Coordinator* as another critical aspect of the position. Over 2/3 of CSEs have had a critical role as *Customer Interface*, *Glue*, and *System Designer* as part of their first CSE position (71%, 71%, and 67%, respectively). One CSE explained that the variety of roles played in the CSE position is somewhat dependent on the organization. For example, when he was a CSE at a small organization, he had to be a "jack of all trades" and therefore played a multitude of roles; now at a much larger organization, his role of CSE is more specialized because there are more people available to perform other roles. Both, he explained, had benefits for development of systems engineering related skills.
- The more detail-oriented roles were less commonly seen as part of the CSE position, but did occur. Often, when a role was not directly performed by the CSE, interviewees explained that the role was simply performed by a member of the team and overseen by the CSE.
- It is fairly uncommon for a CSE to function as an *Instructor/Teacher;* generally, this was explained

because a CSE has a wealth of responsibilities and activities and often does not have time to devote to instruction.

- It is also rare for a CSE to function as a *Program/Project Manager*. In the instances where this did occur, it was generally on a smaller project, where the overall smaller staff required that single individuals take on multiple positions.
- *Organizational Manager* is again a very uncommon role for a first CSE position, as the focus for CSEs is generally technical over administrative.
- None of the CSEs had detailed design responsibilities in their initial CSE roles because the CSE must oversee a team of systems engineers and other engineers. Occasionally a CSE would play the role of subject matter expert or detailed designer in a CSE position, but this always occurred much later in their careers, often as part of a smaller project or a proposal.

The roles played in a first CSE position would tend to further proficiencies in *System's Domain and Operational Context*, *Systems Engineering Discipline, Systems Engineering Mindset, Interpersonal Skills*, and *Technical Leadership.* Though a CSE may not lose skills in *Math/Science/General Engineering*, however, because the role(s) played seldom focus on these areas, it would be unlikely for a CSE to gain proficiency in these areas.

Roles Played throughout CSEs' Careers

Figure 41 and Figure 42 provided insight into the roles played by CSEs up to and including their first CSE positions. Figure 43 provides an overview of the roles played by CSEs throughout their careers, spanning up to their participation in Helix.

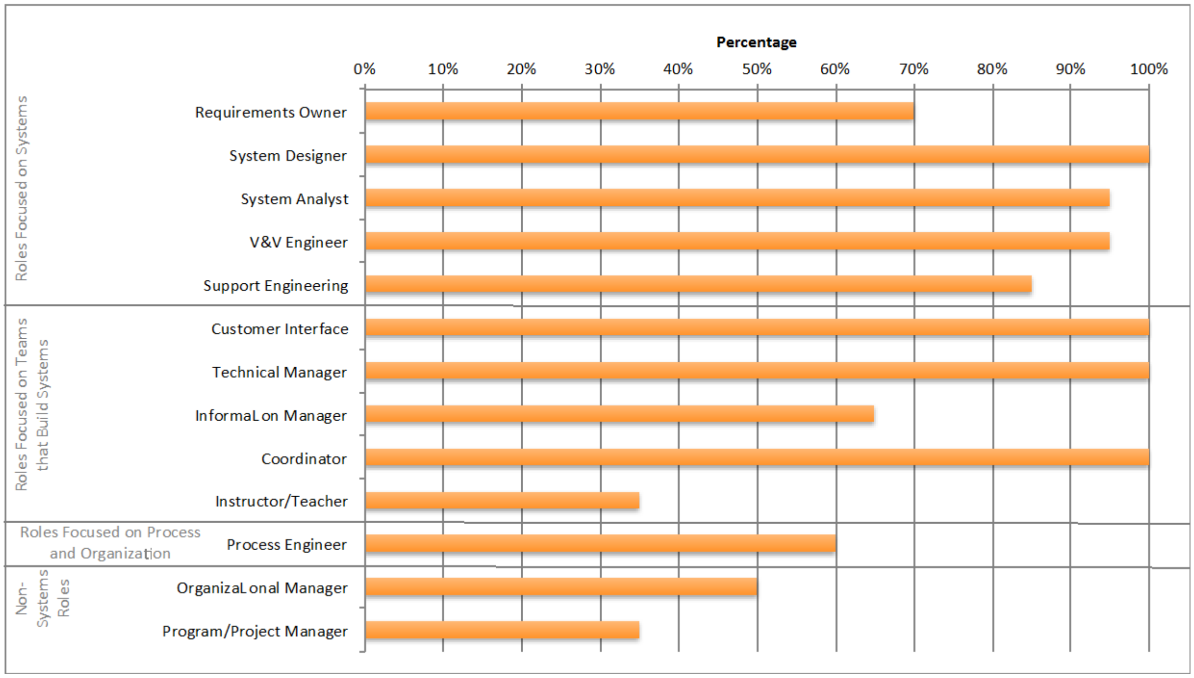


Figure 43. Roles Played by CSEs Throughout Their Whole Careers

The following observations can be made from Figure 43:

- All CSEs in the Helix sample have played the roles of *System Designer*, *Glue*, *Customer Interface*, *Technical Manager*, and *Coordinator*.
- Architecture skills are associated with proficiencies in *Math/Science/General Engineering*, *Systems Engineering Discipline*, *Interpersonal Skills*, and *Technical Leadership*. This provides further insight into the population of CSEs, as 18% have also been the chief architect for a system.
- Even in areas where less than 100% of CSEs have played a role, a much higher percentage of CSEs have played the role than seen in the general Helix sample. For example, 87% of CSEs have played the role of *Requirements Owner* while in the overall Helix sample, this number is just over 40%.

It is clear that the percentage of CSEs who have played these roles continues to rise even after their first CSE position and overall, the percentage of CSEs who have played these roles is considerably higher than in the general Helix population. This indicates that CSEs overall have experiences playing a wider range of roles and that this broad variety of roles continues throughout their careers; it does not stop when they earn the title of “Chief Systems Engineer”.

5.2 INSIGHTS FROM INCOSE SEP ANALYSIS

Among the three levels of INCOSE SEP certification, ESEP is the highest category. ESEP applicants are required to submit information of twenty or more years of work history relevant to systems engineering, and are therefore expected to possess significant experience in systems engineering. In the seniority levels defined by Helix, ESEPs that are successfully certified are Senior systems engineers.

As discussed above, analyzing the career paths of CSEs provides some insights for career development. Similarly, within the INCOSE SEP data from certified ESEP applicants, a subset of those who have held a CSE position or a position equivalent to CSE was analyzed. A category called 'ChiefX' was identified, that included certified ESEPs who have held CSE titles or other equivalent titles, specifically, Chief Engineer, Chief Architect, Chief Systems Architect, Chief Principal Engineer, and Chief of Systems Engineering. A comparison of CSEs from the Helix interview data and ChiefXs from INCOSE SEP data was used to partially validate the Helix sample against a larger diverse international sample of systems engineers.

5.2.1 EDUCATION OF CHIEFXS

Table 14 compares the highest degrees obtained by CSEs in the Helix interview data and ChiefXs from the INCOSE SEP data. The highest degrees compare well between the two samples, except that ChiefXs have almost twice the percentage of doctorate degrees than do CSEs.

Table 14. Highest Degree Attained for CSEs (Helix interviewees) and ChiefXs (ESEPs)

Degree Level	CSEs	ChiefXs
Associate's	0%	0%
Bachelor's	23%	23%
Master's	64%	60%
Doctorate	9%	17%

Figure 44 compares the bachelor's degree majors of CSEs and ChiefXs. Electrical engineering comes out as the most popular major in both samples. Though there are some variations, mechanical engineering, computer engineering / science are the next popular majors for ChiefXs. There are some majors such as civil engineering, aerospace or aeronautical engineering, and industrial engineering that are found in one sample but not in the other.

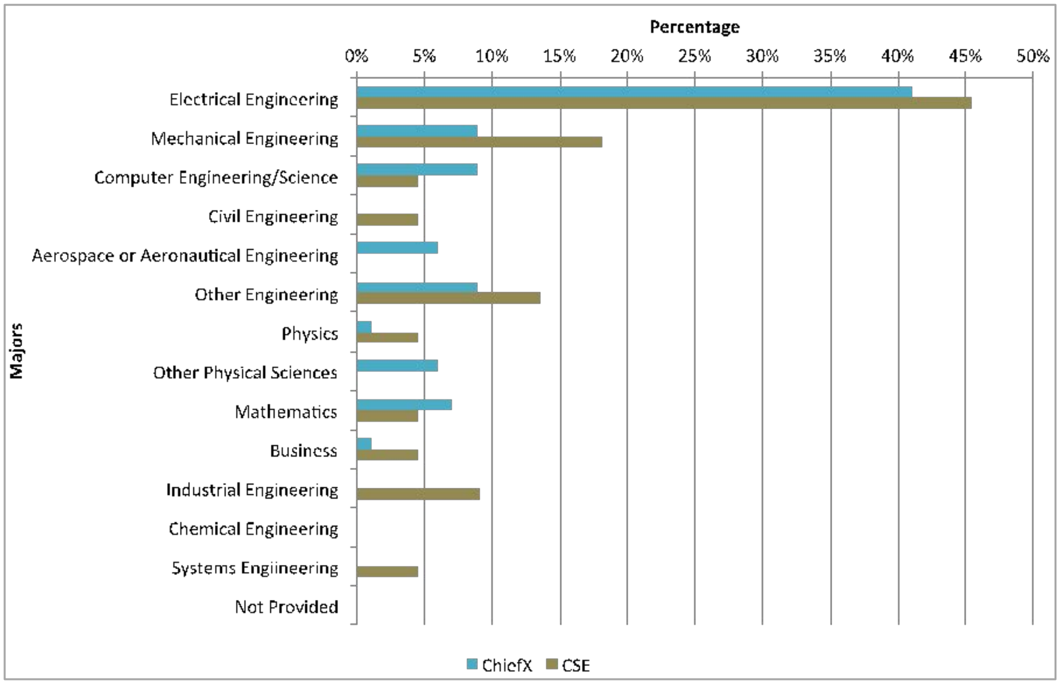


Figure 44. Comparison of Bachelor's Degree Majors between CSEs and ChiefXs

Figure 45 compares the master's degree majors of CSEs and ChiefXs. The most prevalent master's degree major attained was in the area of business – 38% of the CSEs and 39% of the ChiefXs sought a management master's. Most frequently this was an MBA or other management variant. A little less than a quarter of CSEs and ChiefXs pursued a master's in electrical engineering. Almost 10% of CSEs and 13% of ChiefXs complete a master's in systems engineering. The trends observed in master's degree majors indicate similar education profiles for both the CSEs and the ChiefXs.

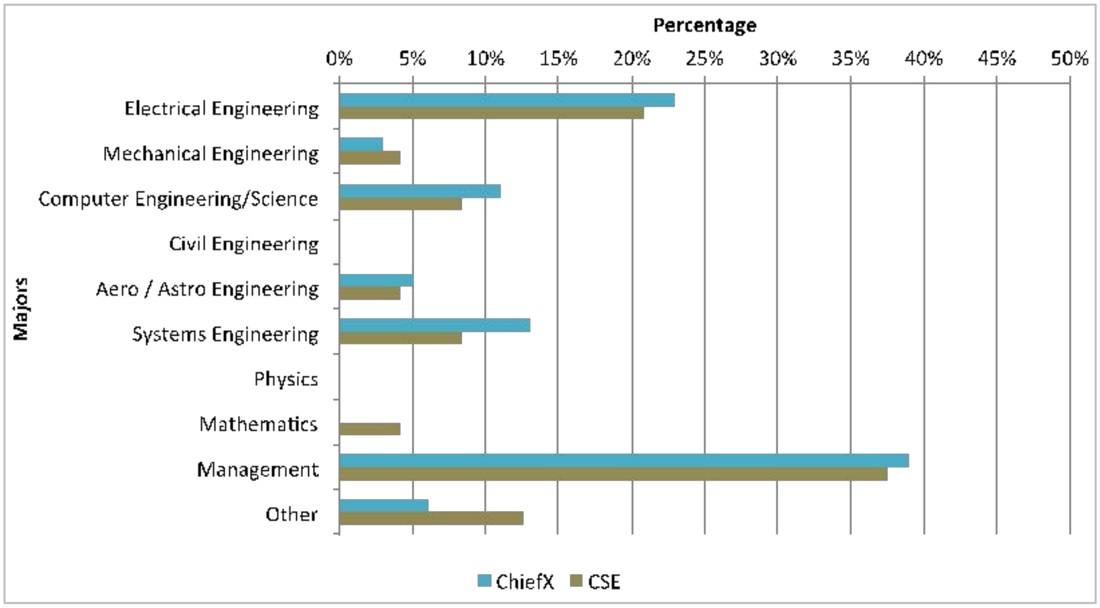


Figure 45. Comparison of Master's Degree Majors between CSEs and ChiefXs

As shown in Table 14, 17% of the ChiefXs have doctorate degrees. This is much greater than in the Helix sample, of which 9% of CSEs have doctorate degrees. Similar to findings from the Helix sample, there was minimal convergence in any specific academic discipline for doctoral study. Of the ChiefXs, 13% of the applicants sought a doctor of philosophy in engineering, mechanical engineering, computer science and systems engineering/integration; the other 4% includes one-off doctorate degrees such as Applied Mechanics and Juris Doctor.

5.2.2 Career Roles played by Certified ESEPs and ChiefXs

Helix identified the *Atlas* roles among INCOSE ESEP applicants, using text-based searches. Figure 46 compares the roles played by senior systems engineers and CSEs from the Helix interview data, compares the roles played by certified ESEPs (non ChiefXs) and ChiefXs from the INCOSE SEP data.

Note that the roles reflected in this section align with *Atlas 0.5*, not the roles framework presented in *Atlas 1.0*. The reason for this is that, though the Helix analyses have been updated as much as possible to reflect the updated framework, the team has not yet had an opportunity to review the INCOSE CSEP data to update this analysis. The only way to provide comparable insights is to use the same framework (reflecting *Atlas 0.5*) for both.

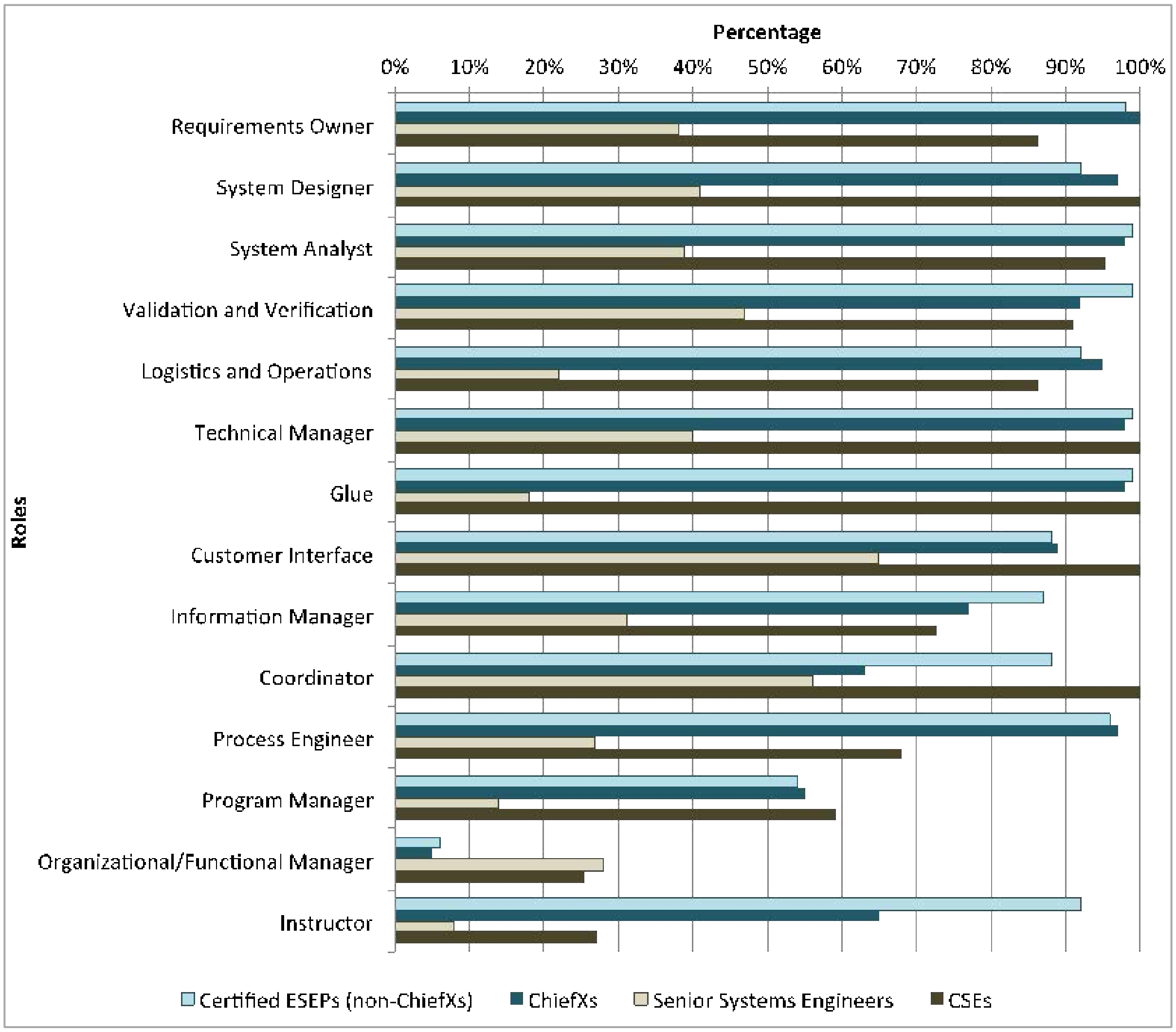


Figure 46. Comparison of Roles Played Throughout Career between Helix Interview Data and INCOSE SEP Data

The career roles played by the INCOSE certified ESEPs and ChiefXs most closely resemble the roles played by the CSEs in the Helix sample. Since INCOSE has an extremely selective process for certifying ESEPs, the individuals that receive the certification have *Experiences* equivalent to the CSEs in the Helix sample rather than the senior systems engineers. Therefore, the senior systems engineers, while they are more seasoned than junior or mid-level systems engineers, do not, in general, have the breadth and depth of *Experiences* in different roles as compared to the CSEs, the ESEPs (non-ChiefXs), and the ChiefXs.

Additional insights include:

- The *Organizational/Functional Manager* role is more common in the Helix interview sample. This could imply that either those who apply for ESEP are more likely to be in a technical track and therefore not have the experience in management to take on this type of role; or ESEP applicants omitted reporting such roles since the application form solicited only systems engineering related tasks and functions.
- The Helix sample identified the most seasoned systems engineers who hold critical systems positions are only occasionally instructors at some point in their career. But in the ESEP applications, the role of instructor is called out more explicitly, and applicants are requested to provide details on training. This may have led to a higher reporting rate among ESEPs than CSEs among Helix interviewees for the role *Instructor*.
- The Helix sample showed that systems engineers often play multiple roles while they are in a single position. This finding was mirrored in the ChiefX subset. More than half of the ChiefXs had compound titles, and therefore it was assumed they were actively performing multiple roles. The most frequent keywords in ChiefX compound titles indicate that organizational leadership is typically a complementary position to CSE. Three of the most frequent keywords in ChiefX compound titles were Manager (17%), Lead (12%), and Director/Head (11%). The other frequent keyword, Architect (8%) indicates that CSEs hold technical and system-specific roles.
- For all ChiefXs who held 3 or more ChiefX titles at some point in their careers (15% of the ChiefX subset), each individual progressed into larger and more complex systems. The Helix sample showed that one common career path for systems engineers' stems from a highly technical position, and through the growth of their careers, the systems engineers take on more responsibility and leadership roles, which correlates with growing of interpersonal skills.

5.2.3 FIRST CHIEFX POSITION ROLES

Figure 47 provides a comparison of the roles played by ChiefXs in their first ChiefX position and roles played by CSEs in their first CSE position from the Helix interviewee dataset.

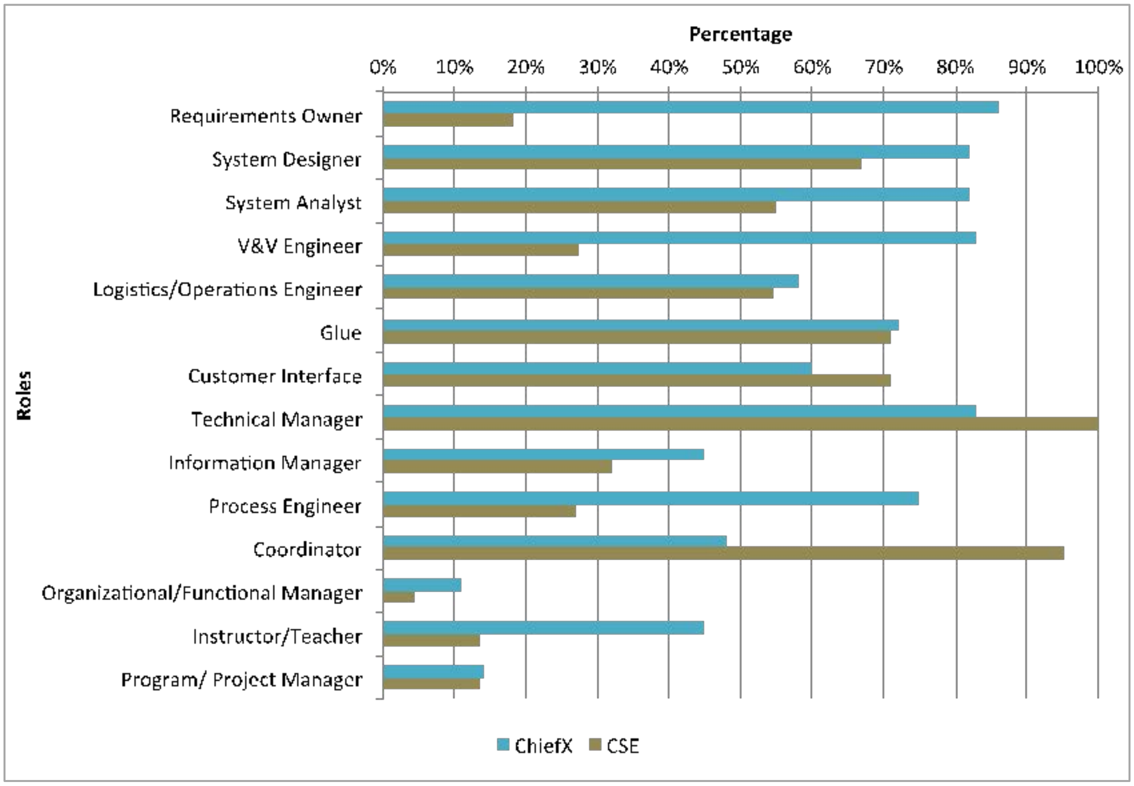


Figure 47. Roles Played in First ChiefX and CSE Positions

As is seen in Figure 47,

- The roles played most commonly during the first ChiefX position are *Requirements Owner*, *Technical Manager*, *Validation and Verification Engineer*, *System Designer*, and *System Analyst*. Conversely, the most frequent roles in the first CSE position are *Technical Manager*, *Coordinator*, *Glue* and *Customer Interface*.
- The CSE sample experienced technical management and programmatic aspects of systems rather than the more technical system lifecycle roles that were experienced by the ChiefXs.
- Almost half (45%) of the ChiefXs played ten or more different roles in their first ChiefX position - 35% played between 5 and 9 roles (inclusive), and 20% played less than 5 roles. Only the *Program Manager* and *Organizational/Functional Manager* roles are performed by less than 40% of the ChiefXs.
- Almost half of the ChiefXs played the role of *Instructor/Teacher* while in their first ChiefX position. This indicates the knowledge base of these individuals, and their willingness to share their critical understanding and experiences with others, ultimately leading to an improvement of their organization.

5.2.4 VALUE OF INCOSE SEP ANALYSIS FOR ATLAS

The INCOSE data was an excellent benchmark for the Helix sample. Characteristics and patterns identified in *Atlas* were further evidenced via comparison of certified ESEPs and ChiefXs with the senior systems engineers and CSEs from the Helix interview sample. There were no major discrepancies that would indicate a need for reassessment of *Atlas*. INCOSE SEP applications provided a wealth of data for use in identifying the *Educatio*n and *Experience* backgrounds of those who have been systems engineers in industry for over 20 years, and are certified as knowledgeable, experienced and accomplished professionals in systems engineering.

5.3 A SAMPLE CAREER PATH OF A CHIEF SYSTEMS ENGINEER

The position of a Chief Systems Engineer (CSE), or equivalent, is considered to be one of the highest technical positions for a systems engineer, and a framework visualizing a career path was presented in Section 5. The career path of one particular CSE from the Helix data set, 'Athena' (a fictional name) is visualized below in Figure 48.

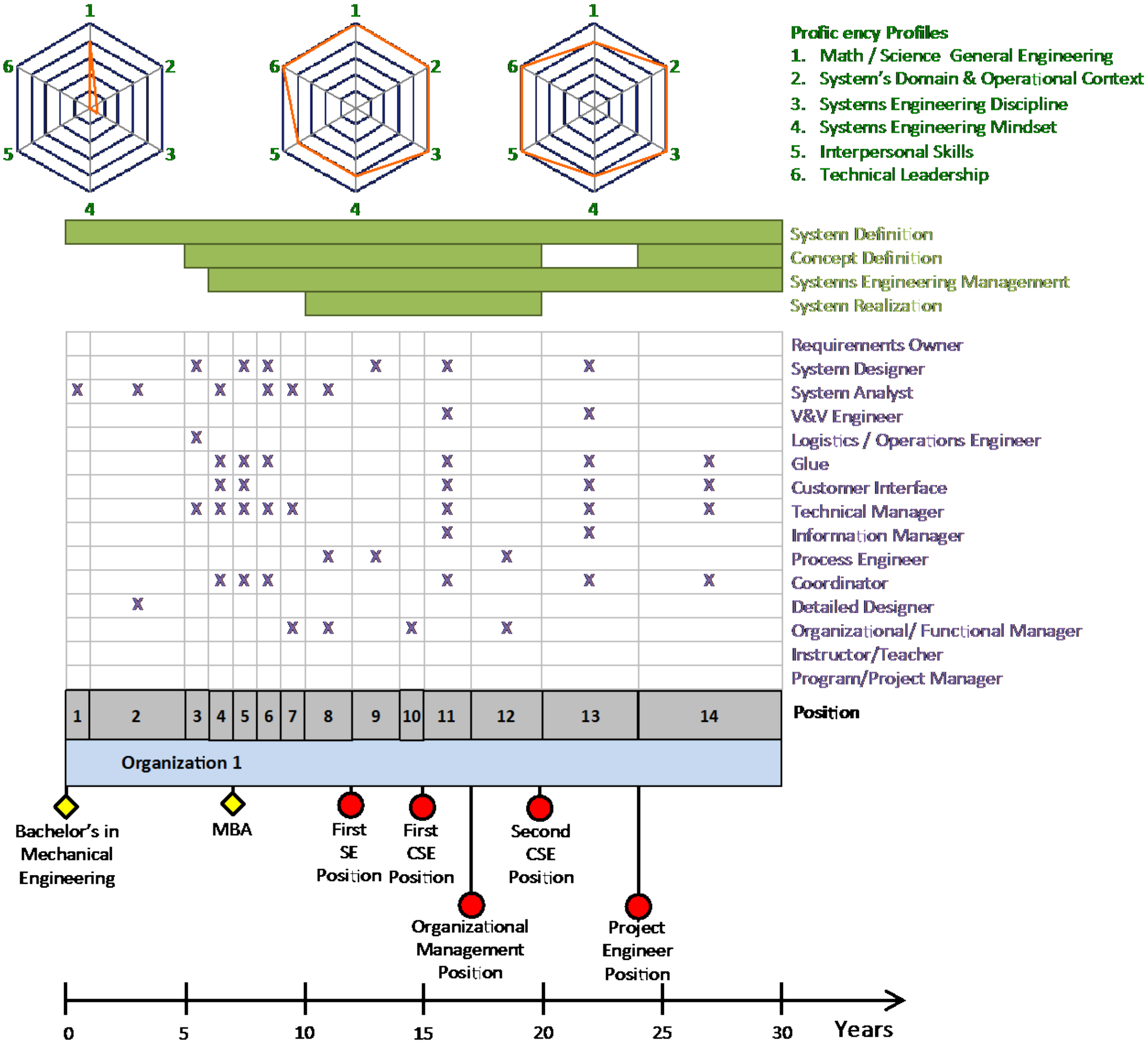


Figure 48. A Sample Career Path

This visualization provides a great deal of insight into Athena's career path. First, it provides detail on Athena's education – a bachelor's degree in mechanical engineering achieved before entering the workforce and an MBA received 7 years into her career. It shows that Athena had a career length of 30 relevant years at the time of her participation in Helix, and Athena had worked in only one organization throughout her career. Within that organization, she had held 14 different positions for varying lengths of time. In terms of key positions, the first formal "systems engineering" position occurred about 12 years into her career and because she is a CSE, the timing of her first CSE position – 15 years – is important as well. Following her first CSE role, Athena became an organizational manager for a few years before taking a second CSE position and then becoming a project engineer. Over half of her career has been spent in formal systems engineering positions and nearly half in senior technical positions (CSE and project engineer).

Above each position, the roles played in that position are detailed; these were identified based upon the description of the positions provided by Athena. These provide further insight not only into what activities Athena was performing at different points in her career but also into the proficiencies she was likely to be developing.

The leftmost proficiency diagram in Figure 48 shows projected proficiency after completion of Athena's bachelor's degree and during her first position. The role played by Athena in her first position was *System Analyst* – specifically applying mathematics, physics, and engineering principles to a real system. The proficiency level for *Math/Science/General Engineering*, therefore, is projected to be higher than what was gained in an undergraduate program. In Position 1, Athena gained *Experiences* working on a specific system within a specific domain, gaining moderate proficiency working in the *System's Domain and Operational Context* proficiency area. She was also exposed to the performance of engineering and tangentially became familiar with the concept of systems engineering and how a system analyst can support systems engineering activities, gaining a moderate proficiency in the *Systems Engineering Discipline* area.

The middle proficiency diagram in Figure 48 shows projected proficiency of Athena going into her first CSE position. By analyzing Athena's first ten positions for the roles played, lifecycle exposure, domain exposure, and types of systems, the projected proficiency would be expected to increase considerably. In fact, going into her first CSE position, Athena would be expected to have "high" proficiency in all areas except *Systems Engineering Mindset* and *Interpersonal Skills*. This is based, again, on the information available on her career path and does not take into account any "baseline" proficiencies. If, for example, Athena was naturally very strong in *Communication* or tended toward *Big-Picture Thinking*, she may have "high" proficiency in these areas, but Figure 48 provides a projection based on what she has had the opportunity to learn based on her *Education* and *Experiences*.

Finally, the rightmost proficiency diagram shows that with additional responsibilities and *Experiences*, Athena maintained her proficiencies. One area that may be surprising is the decrease in proficiency in *Math/Science/General Engineering*. This is because as Athena took on increasing leadership responsibilities, over time she performed less of the detailed engineering or analyses. These skills became less prominent, though were certainly not lost completely. Interestingly, this was a common pattern seen in senior systems engineers. One senior systems engineer explained that he did not need to do detailed engineering or calculations at his level, but he needs to know who can do that work and have a "gut feeling" for whether the answers he receives from those individuals were sensible.

As demonstrated above, the detail provided by analyzing the different elements of a career path provides considerable into proficiency. The type of simple visualization shown in Figure 48 can be created for any combination of the elements of a career path.

6 *ATLAS* DEPLOYMENT – A GLIMPSE

In 2015 and 2016, the Helix team has worked with five organizations that are interested in implementing the findings from *Atlas 1.0*. Based on these efforts, the Helix team has developed use cases as well as considerations for individuals and organizations wishing to implement *Atlas*. The considerations for using *Atlas* in Section 6.2 (below) are incorporated throughout this document, but are collected here to make them more easily accessible.

6.1 USES AND USE CASES

There are two primary ways in which *Atlas* can be used – to provide insight and guidance to individuals and to inform organizational-level efforts. Guidance on how to use various aspects of *Atlas* is provided throughout the various sections of this document. This section pulls this together, describing at a high level the major expected uses for *Atlas*. Several organizations have, to varying degrees, tried all of them. Figure 49 shows individual uses.

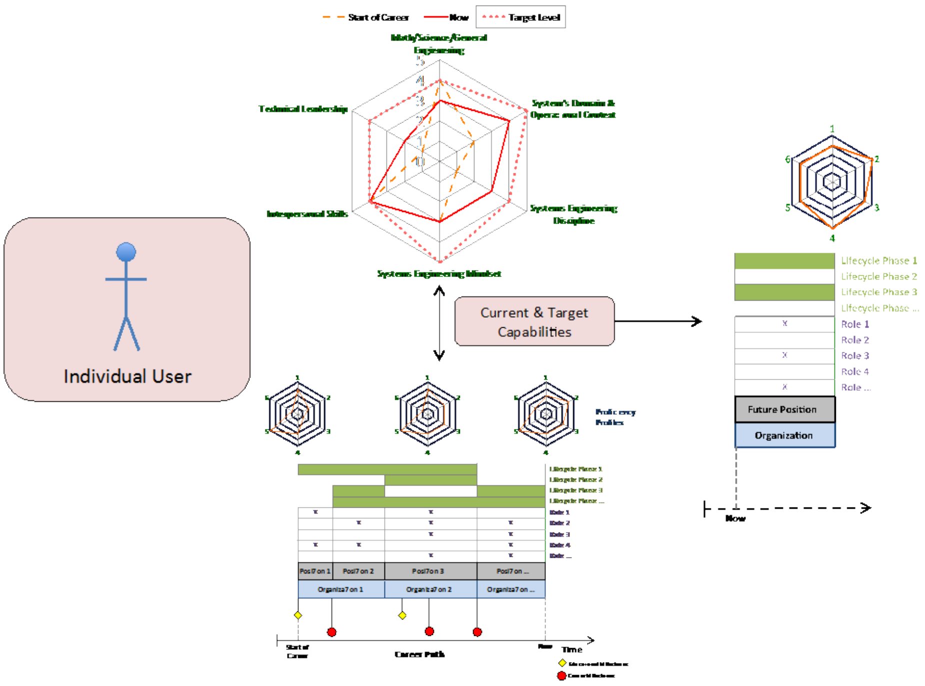


Figure 49. Expected Uses for an Individual User

As shown in Figure 49 an individual is expected to be able to:

1. **Use Proficiency Self-Assessment to identify current proficiency levels as well as past trends**. As described 5.8.1, proficiency profiles are most effective when they are examined over time. An individual will benefit from understanding these patterns and using them to inform potential targets for the future.
2. **Use Career Path self-assessment to categorize and analyze past forces** (experiences, mentoring, and education and training). This data can be used to identify any clear gaps in Forces over time.
3. **Use Proficiency and Career Path self-assessments to identify a way ahead for a career.**
    - **Identify a target state**. Proficiency profiles provide a useful starting point for discussions

with the organization about potential future positions – what positions make sense, what the proficiency expectation for this position are, etc. These future goals could be based on known positions within an organization (e.g. "I want to be a systems architect") or individual desire (e.g. "I am interested in this type of system"). Target states can often be clarified in discussion with a mentor or leader who understands the expectations for different types of positions in the organization as well as the individual's proficiencies.

- **Assess gaps between current and target proficiency.** As illustrated in Section 5.8.1., once target proficiencies have been identified, they can be plotted in a proficiency profile along with current proficiency levels. This provides an easy way to visualize gaps between current and target proficiency, helping an individual understand where they need to focus their growth.
- **Pair proficiency gaps with career path information to identify potential ways to improve proficiency.** Experiences, mentoring, education, or training are all ways that proficiencies can be improved and often a combination of forces is required to reach a target proficiency. For example, a gap in systems engineering discipline may initially be addressed by targeted training or education programs. However, a best practice identified by Helix is that this must be applied on the job immediately in order for any improvements in proficiency to become permanent. If a mentor can help guide the application of new learning in these experiences, there is likely to be additional improvement in proficiency as well. All of these considerations provide a starting point for planning and can be used to discuss possibilities with management or leadership.

Figure 50 shows expected organizational uses of *Atlas*.

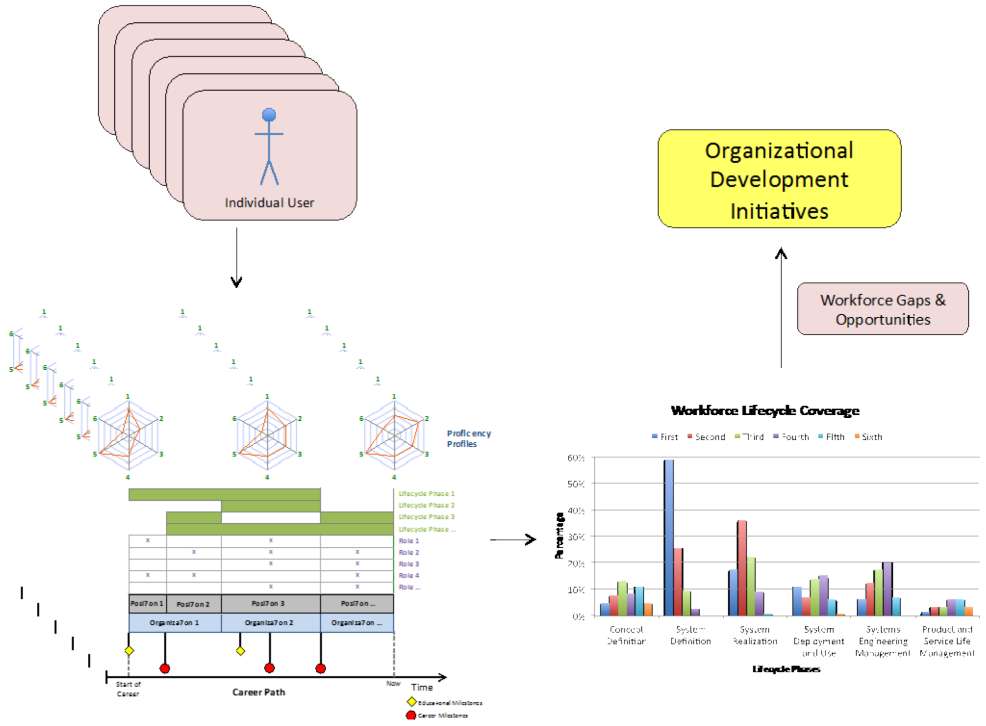


Figure 50. Expected Uses for Organizations

As shown in Figure 50, an organization is expected to be able to:

- **Treat the workforce as a collection of individuals.** Each individual can gain insight on current and potential target capabilities as discussed above. By taking the proficiency profiles – current and target – for a group of individuals, the organization can gain insight into any current capability gaps and understand desired future capabilities. For example, if no one in the group has higher than a proficiency level "6" in Technical Leadership, but the organization feels it needs several individuals with a level "8" proficiency or higher, then the organization has identified a critical skills gap. Paired with the target states, the organization can then identified individuals who are already interested in developing their Technical Leadership skills and can focus opportunities related to technical leadership on these individuals. Likewise, the organization may identify individuals who are believed to be "high potential" for technical leadership who may not have identified this in themselves and enable a conversation about future directions.
- **Use the career path data from individuals to identify patterns of the overall workforce.** Similar to the point above, organizations can use the career path data for the individuals in the workforce to identify overall patterns. For example, perhaps less than 5% of the workforce has experience in the role of "Concept Creator". If the organization has identified this as a critical area for growth of systems engineers, this may indicate that the organization should develop initiatives to foster growth in this area. Likewise, if there is an area of the lifecycle that is commonly missed in the workforce, the organization can determine if this is a critical gap or whether it makes sense in the organizational context. For example, if only 10% of the workforce has experiences in "Systems Deployment and Use", but the organization does not participate in operation and maintenance of its systems, then this may be seen as acceptable. The organization also now has data about the workforce that it can use to fill gaps. For example, if the organization needed perspective on a project specific to "Systems Deployment and Use", the data will provide insight on who in the organization has this experience.
- Use workforce data to improve or create new organizational development initiatives. Using the gap analysis across current and future desired capabilities, the organization can identify opportunities or set strategic goals regarding workforce capability. As illustrated in the examples above, this information would then provide opportunities for improved or new development initiatives.

6.2 RECOMMENDATIONS FROM HELIX

Senior and mid-level systems engineers were asked to recommend what steps younger systems engineers should take to grow and become more effective. Junior systems engineers were asked what they believed they needed to do personally in order to grow and become more effective. A total of 167 different recommendations have been catalogued by the Helix team, which validated the tenets of *Atlas* and other findings throughout this report. They have been grouped into a series of primary recommendations below, with discussion for the role of individuals and organizations as appropriate for each. Percentages reported below are the percent of total recommendations that address this issue. Note that the percentages do not add up to 100% because some recommendations may touch on multiple areas or address the balance between different areas (for example, the balance between depth and breadth of experiences.)

- **Systems engineers should seek broadening assignments**. Nearly half of the recommendations (47%) focused around the need for individuals to have a wide variety of *Experiences*.
  - *Individuals* should seek opportunities to explore new areas of the lifecycle, perform new roles, become better acquainted with the technologies of the domains, have additional leadership opportunities, or become more familiar with different aspects of the organization. This may be done in the context of formal rotational programs in an organization, or can be done more informally. Most systems engineers generally focused on the need for breadth of *Experiences*, but some recommendations (13%) also focused on the need to balance depth and breadth of *Experiences*. Conversation around depth generally indicated that an individual be able to understand the technologies, components, and interfaces of the system they are working on; have an understanding of the technical work required to create that system; and have enough depth to be able to have good conversations with and earn the respect of other engineers that support that system.
  - The CSEs in the Helix sample all had *Experiences* in four or five of the six areas of the systems lifecycle, all of them experiencing *System Definition*, *System Realization*, and *Systems Engineering Management*. In addition, CSEs had *Experiences* in 11-13 of the identified systems engineering relevant roles. As a general guideline then, systems engineers should seek roles that provide exposure to new aspects of the systems lifecycle or new roles. Another fairly common pattern was to balance breadth in these areas with depth in a single domain or technology area.
  - *Organizations* should enable individuals to obtain these broadening assignments. It was fairly common for organizations to do this through formal rotational programs. However, concerns were expressed about the structure of rotational programs and the availability of these programs to different parts of the workforce. In general, these programs were only open to younger systems engineers who were new to an organization. In addition, there were concerns about the approach for some programs. In particular, senior systems engineers reported that junior systems engineers who complete these programs are often given the message that they will be leaders of large systems quickly, but due to the lack of availability of these more senior positions and the fact that these individuals are relatively inexperienced, this does not often happen. This can lead to frustration for many junior systems engineers and some will choose to leave the organization.
  - Another option, then, is that organizations can instead create a culture that is more supportive of allowing individuals to move informally between positions to gain this

breadth of experiences. In most organizations, this was reported as very dependent on the manager of each individual systems engineer. If organizations agree that this is important, then they should put in place policies that make transitions between positions easier and focus on the overall benefits to the organization of having more effective systems engineers. This will help remove some of the stigma that many junior systems engineers felt was associated with moving between different parts of the organization.

- **Systems engineers should engage in mentoring relationships**. *Mentoring* can play a critical role in helping systems engineers to improve their proficiency.
    - *Individuals* should seek out mentoring relationships with more seasoned systems engineers throughout their careers and regularly engage their mentor in discussions and seek feedback. Likewise, individuals should look for less experienced systems engineers to connect with and, where possible, provide guidance and insight based upon their own experiences.
    - *Organizations* should seek to foster the exchanges generated by mentoring relationships. The Helix data shows that rigid formal mentoring relationships are often less effective than informal ones. However, organizations can encourage mentoring relationships by providing guidance, training, or other resources on how to work in a mentoring relationship; set expectations that senior systems engineers should in general function as mentors; or make engagement in mentoring relationships part of how systems engineers are evaluated.
- **Systems engineers should ask probing questions**. Though a smaller percentage of systems engineers spoke directly about this when asked for recommendations to individuals (8%), this was a theme that emerged in other aspects of the interviews well.
    - *Individuals*, particularly junior systems engineers, should “constantly be asking the ‘why’ questions.” This refers to both technical questions about the system, questions about the decisions made in the past on a system, and about how the organization approaches systems engineering work in general. Senior systems engineers said that often junior systems engineers do not seem empowered to ask these questions, but *they are critically important to developing a big-picture view of both the system and the context of systems development within the organization*. It is important, then, that more senior systems engineers also approach these questions not as challenges to themselves or their authority, but instead as opportunities to help junior systems engineers improve their overall understanding.
    - *Organizations* should encourage a culture where this type of questioning is seen as a benefit and not a challenge to authority. In addition, organizations could develop some guidelines or history of why business or engineering is done in a certain way. One senior systems engineer stated, “I’m not sure we necessarily give young engineers training and background to say, ‘Culturally why our processes are the way they are today is because historically for the last 25 years prior to the last 10 years, that we followed these regimented standards for how to do development efforts to be successful with our customers.’ And if we did that more, I think they would understand where they are in the process; they would understand why we do what we do….” This is the type of exchange that could also be encouraged as part of a mentoring relationship.
- **Systems engineers should take advantage of *Education & Training* opportunities** and look for

ways to immediately implement their learning in practice. There were a number of systems engineers who believed that *Education & Training* play an important role in development and provided specific recommendations on how to best utilize these growth opportunities (22%). While the details of available training differed between organizations, there were some common themes.

- *Individuals* should make themselves familiar with the *Education & Training* opportunities offered within an organization and become their own advocates for participating in those programs. In addition, 16% of those who spoke about *Education* specifically highlighted that it is critical for individuals to immediately apply on the job what they learn from *Education & Training*. For example, individuals of the same seniority, working in the same unit of an organization, would often have conflicting views on the usefulness of the same training courses. When asked their rationale, individuals who were able to apply the techniques on the job were far more likely to find the training useful and effective than individuals who felt it did not readily apply to the work they did. Several individuals indicated that in order for the training to become useful, they would have to retake it and then immediately apply it on the job.

- Though training opportunities varies widely between organizations, graduate education was common and valued in the sample. The most common and commonly recommended master's degrees for systems engineers were a master's degree in systems engineering or an MBA (or other business-related degree). The master's degree in systems engineering provides a holistic overview of the discipline and was valued as a means to improve the *Big-Picture Thinking* for systems engineers, particularly in organizations where rotational assignments were difficult to obtain or where long life cycles made it difficult for systems engineers to gain a more holistic understanding hands-on. MBA degrees provided the business context for many of the engineering decisions, helping individuals to understand how to make technical issues more understandable and important to their organizations.

- *Organizations* should set policies in place that make it easier for individuals to take advantage of available *Education & Training* opportunities. A common roadblock heard when discussing *Education & Training* was that the organization did not provide sufficient support for training; for example, a course deemed critical may be offered only once a year to a limited number of individuals or managers would consistently decide that individuals could not have time away from their work to attend courses. While this is an incentive for individuals to become their own advocates, it also highlights the role of the organization to provide adequate opportunities for and support of training initiatives. The costs associated with training were also commonly discussed as issues, with the burdens being felt by individual departments, again limiting opportunities. One senior systems engineer recommended that organizations "can pay overhead costs - in other words, the cost of the trainer to be here; you could pay that off a top-level budget. You don't need to roll that out to individual departmental budgets, which then becomes the purview of the individual managers saying 'well you went to that class last year, so you can't go to one this year because you've used up your budget for two years'."

- Finally, perhaps the most important thing that organizations can do is take a more strategic approach to *Education & Training*, and particularly matching these opportunities to the ability of individuals to apply what they learn in practice. While individuals should look for opportunities to apply what they gain from education and training, organizations can also provide a better approach to identifying which individuals should attend training based on their ability to apply the concepts in practice, which would improve overall retention and improve the effectiveness of these programs.

- ***Systems engineers must mind their own careers***. Some organizations do have clear career paths for systems engineers, but most do not. Though only 5% of specific recommendations for individuals focused on this topic, it also came across strongly in the mentoring analysis.
    - *Individuals* should take responsibility for their own careers, seeking opportunities for growth and planning next steps. This was also discussed throughout many different aspects of the interviews. Because most organizations do not have clear career guidance for systems engineers, it is necessary for individuals to take responsibility for their own careers. *Mentoring* was often discussed as a useful way for individuals to explore options and choose appropriate next steps.
    - *Organizations* should create basic career guidance for systems engineers. Of the organization-specific recommendations, 11% focused specifically on the lack of guidance available to systems engineers when planning their careers. This guidance could include the types of roles individuals should expect to play at certain stages in their careers and when certain types of training may be most appropriate. Even though individuals should take responsibility for their careers, several senior systems engineers recommended that if organizations could create general principles for systems engineers to follow in their careers, it would greatly support junior systems engineers in making better decisions about which positions to seek.

6.3 BEST PRACTICES

This section collects the various best practices identified throughout the document. These are also described in each relevant section, but are collected here for easier reference.

6.3.1 BEST PRACTICES FOR PROFICIENCY

It is important to note that to date, though there is a rubric to guide proficiency self assessment, the numbers used for proficiency self-assessment are more comparative than strictly quantitative; i.e., they are useful to understand how an individual views their changes in proficiency over time. In a group setting during interviews, participants were able to discuss what they meant by their proficiencies and by comparing these with those of their peers, make adjustments so that within a single group, the numbers became more consistent.

However, it would be unwise to over-interpret the numbers used for proficiency assessment, particularly when comparing proficiency levels across a number of individuals. Proficiency profiles provide a very useful snapshot of an individual's perceived capabilities. But the greater value to date is the *conversation* around the rationale for self-assessment. One organization that has been working to implement *Atlas* uses the proficiency in exactly this way – as a prompt for discussion between individuals and their leaders to inform career planning. The proficiency profile, then, becomes a tool that can be used to validate perceived capabilities with peers or managers and this validated view can then be used to support future planning.

In 2017, the Helix team plans to use implementation opportunities as a way to further refine the rubrics for proficiency levels so that they become more consistent and comparable.

6.3.2 CRITICAL FACTORS FOR MENTORING

There are several factors that significantly influence the success or failure of any mentoring arrangement, and interviewees provided a number of anecdotes and personal experiences to elaborate on this. Recognizing and addressing these factors by the organization and the individuals involved could vastly increase the chances of success with a mentoring arrangement.

- **Mentoring Has Its Limitations**: Mentoring does not always work in all organizations, for all employees; mentoring does not work in isolation, but in conjunction with other organizational initiatives to support its employees, such as training and education; and mentoring cannot be solely relied on for knowledge capture from a retiring senior engineer. Even with respect to mentees who stand to benefit the most in a mentoring arrangement, one interviewee stated, "I do not see different traits among those formally mentored, informally mentored, and not mentored at all." While there are many other cases that would prove this statement wrong, it only shows that mentoring is not a universal solution. One interviewee stated "I've seen great systems people that have no formal mentoring". While it is possible that any mentoring could have been implicit and not visible outside, it shows that while mentoring is highly desired, it may not be essential for everyone. Another interviewee supported this view when he said "I'm proof that [mentoring] is not essential; it just made it more uncomfortable for me."
- **Right Choice, Right Pairing**: Ineffective mentors exist, and just the fact that someone is a true expert with many years of experience would not automatically make them a great mentor. In some mentors 'communication' could be a factor, but 'willingness' could also be a factor. As one interviewee elaborated, "some are good with sharing their knowledge, and there are people who don't want to share any of their knowledge". Similarly, not everyone benefits from being a

mentee; some are more comfortable learning things on their own than from a mentor. Finally, it is most important to pair the right mentor with the right mentee – alignment between them is important; "[mentoring] is a two way street" said one interviewee, and it can become a "life long relationship".

- **Balance between Formal and Informal Mentoring**: While most interviewees had a personal preference for informal mentoring, everyone agreed that a bit of formality does help. Therefore, establishing the right balance between formal and informal mentoring by the organization is critical. As discussed in Section 0, there are some aspects of mentoring that benefit from a formal arrangement – such as establishing goals and objectives, and keeping track of mentoring arrangements. But for aspects such as deciding the right mentor-mentee pair; and for driving the frequency and nature of the interactions, it is advantageous to let those be informal.
- **Mentor Training**: Mentors may possess the experience and expertise that mentees stand to benefit from, but some guidance on establishing and sustaining a mentoring relationship would help – it does not come naturally to all mentors. Providing some guidance on how to establish goals and objectives for mentoring, and on how to balance teaching and guiding (for self learning by mentees) would be beneficial to mentors. In one organization, interviewees mentioned that there was a mentoring manual, but it was very dated, and that currently mentoring happens in a more ad-hoc manner.
- **Make Mentoring Visible**: If any mentoring program is officially established or encouraged by an organization, it must also take the effort to make it known to all its employees. In some organizations, the interviewees were not aware if their organization had a mentoring program or not. In some cases, top management initiated a mentoring program that failed to percolate through the organization to reach the potential mentors and mentees. One interviewee said, "I didn't even know we had [a mentoring initiative] until someone gave me a mentee."
- **Mentoring Needs Time**: Benefits of mentoring cannot be reaped instantaneously – in some cases, it is only after a number of years that mentees benefit from the mentoring that they had received early on in their careers. Similarly, mentoring cannot be a last minute activity that a retiring senior engineer is expected to do in the last few months or weeks of employment – any mentoring done at that time is not likely to be very effective. With many other activities taking priority before retirement, there is usually not enough time mentoring. Mentoring cannot be rushed, even from the perspective of the mentee who cannot be expected to "think of all questions [to the mentor] today."
- **Load on Mentor**: Organizations must always be aware of the load that mentoring places on the mentor – they cannot be expected to do everything that they are responsible for and also do mentoring. At the same time, mentoring cannot be a fill-in activity where a mentor is told, "you are not busy this month – so go mentor this person."
- **Back up a Mentoring Program**: In one organization, a formal mentoring program was rolled out where everyone would have a mentor and a mentee. It was formal; but "it wasn't backed by a lot of horse power", as noted by an interviewee. The organization needs to back up a mentoring program with the required budget and time for the mentor and mentee to engage in a relationship – it is dangerous to establish a mentoring program just to claim that one exists. Breaking a mentor-mentee relationship can also be harmful – such break ups could happen for a number of reasons, but if it is something that an organization could prevent, it should.
- **Terminating a Mentoring Arrangement**: Even while a mentoring initiative continues to remain

active, there are situations that may warrant the termination of a mentoring arrangement. When the mentee chooses to leave the organization, it could be a loss to the organization if the mentee had received mentoring from soon-to-be-retiring senior mentor. Similarly, when a senior mentor chooses to leave abruptly, it could affect the knowledge capture that may have been in process through the mentoring arrangement. However in either of these cases, the organization may not be able to do much. But when there is a likelihood of an existing mentoring arrangement to be disrupted due to promotions or internal re-organization of either the mentee or the mentor, the organization must take care to provide sufficient time to the mentor and mentee to terminate the mentoring arrangement in a cordial manner, especially if the mentoring arrangement was successful and valuable to the mentor and the mentee.

6.3.3 BEST PRACTICES FOR TRAINING

- **Training must be "immediately" applied or lost.** Participants indicated training that could be applied on the job during or immediately after training was internalized and remembered much more effectively than training which occurred "in the classroom" only, which was much more likely to need to be repeated. Likewise, in the discussion of proficiency changes over time, individuals indicated that training that was applied resulted in a *long-term* increase in proficiency while simply attending a training course would provide a *temporary* increase in proficiency that would diminish without application. This aligns with findings from the literature about the importance of the "opportunity to perform" (Burke and Hutchins 2007) for training transfer.
- **Access to training should be improved.** Across government and commercial organizations and across a variety of domains, a common issue that was cited was access to training. The most common issue was that even when training was mandatory or when it was optional but cited as a priority within the organization, it was often difficult to gain access to training. There were a variety of reasons, but as participants discussed the issues, they came down to a few common causes: a lack of funding for training; a disconnect between organizational and project priority; and lack of managerial support. Particularly in government organizations or commercial organizations in the defense industry still recovering from the 2012-2013 sequestrations, the lack of funds for training was a very common concern. There were many individuals who stated that while the leadership of the organization encouraged training, project managers balked at allowing individuals leave time to attend even partial-day training. Likewise, several individuals stated that while they would request training, their managers would not support their attendance, though in some instances this was tied to lack of available training funds. A related issue was the dissemination of information about training opportunities within an organization. In other words, individual systems engineers were not always aware of the training courses available, even though the Helix team was informed of these initiatives by the leadership team.
- **Training can be a critical tool to address anticipated risks.** When asked what risks they were concerned about in their organizations, participants' responses included how organizations could become more agile, better incorporate and integrate COTS into design, and obstacles to implementing MBSE. Common themes were the lack of domain knowledge and lack of clear communication of the rationale behind the processes used within organization. The former would be a way to help younger systems engineers who would have to mature more rapidly to replace retiring senior systems engineers. For the latter, there were many training opportunities on process, but a common criticism of them was that while the process *steps* are taught, the rationale behind the process – how it came to be, why and when it might be tailored – was not understood consistently, particularly among junior systems engineers. When an organization takes the time

to understand its current capabilities and plan desired strategic capabilities, critical gaps will emerge. Helix participants believed that improvements to current training or the development of targeted new training could be a critical way to address these issues.

7 FUTURE DIRECTIONS

As the Helix team reflected on the accomplishments through 2016, a few key research gaps were identified for further analysis. The primary focus of the future work will be to close the identified gaps and document them in a way that will enable Helix to achieve the full potential impact within the community.

1. **Helix has largely elaborated what enables individual systems engineers to be effective, but has not yet explained to nearly the same depth what enables *organizations* to be more or less effective in growing an effective systems engineering workforce.**

   The current data has focused largely on individual systems engineers; additional data must be collected to better understand how to characterize the effectiveness of a systems engineering team, workforce, and the organizational approach to SE effectiveness. The team will also incorporate existing frameworks around organizational change and personality assessments to enrich *Atlas*. Breaking away from the grounded theory approach that has guided the project to date, the team will also pursue qualitative data and deep understanding of the social sciences aspect of workforce development. Helix will also work to incorporate findings from other SERC research tasks and to share findings from Helix with research tasks as appropriate.

2. **The current data has not been fully assessed for patterns and trends needed to provide a better understanding of effective system engineer career paths or for the typical position assignments and their relationship to the systems engineering roles.**

   While this has been done for 181 of the individuals in the sample, this work needs to be expanded to encompass the entire sample. Likewise, the team will examine the entire dataset for wider patterns that were either unclear or not seen in the subset. For example, are there benefits to taking a particular path in the systems lifecycle over another path – i.e. is there empirical evidence to support a "cradle-to-grave" path through the systems lifecycle over starting at the end of the lifecycle or jumping around to different aspects of the lifecycle? Are there likewise similar trends for education timing and patterns? Again, some of the early patterns have been identified with the subset (e.g. as reported in Lipizzi et al. 2015 or Jauregui et al. 2016), but this work needs to be expanded upon and carried throughout the dataset.

   Another question is whether there are common differences in career patterns between government and industry organizations or those who have worked in particular domains. These patterns, then, will provide broad guidance to systems engineers on potential approaches to their careers and can be the basis for decision making.

   The team will achieve insight through data mining the transcripts and highlighting patterns. The Helix team hopes that the data mining approaches will confirm assumptions of closely coupled relationships between distinct proficiencies, forces, or characteristics. Another development through further structuring of the data well be the creation of a lexicon, taxonomy, and ontology that will enhance the understanding of systems engineering, systems engineers, and organizations that pursue problems with the use thereof based upon empirical data. The Helix team hopes that this will further inform the community.

3. **Though the Helix team has collected data on the ways that organizations try to improve their systems engineers' proficiency and some basic trends have been identified, this data has not been consistently available and little or no data has been available on the return-on-investment of organizational approaches.**

In order to better enable organizations to utilize Atlas, this area requires further study. To enable an increased understanding of organizational effectiveness, the Helix team needs to create a robust model(s) that can aid understanding of not only the effectiveness of individual systems engineers but systems engineers as a group, including teams or an organization's overall workforce. As the team builds upon the organizational aspects for nurturing and growing their SE workforce, Helix will be able to incorporate the initiatives and characteristics into their models to showcase interplay between an individual, a team, and an environment. Social sciences will also be essential in providing insights to teaming, synergies related to proficiencies in teams, etc.

4. **In order to achieve a better understanding of the workforce across an entire organization, a more rigorous set of tools that can better support consistent assessments is required.**

   The Helix team will begin with the tools developed to support *Atlas 1.0* and through implementation exercises with multiple organizations, will develop a web-based tool that can be used more broadly and which can be more easily tailored to the appropriate organizational context. It is hoped that data collected using this tool can be appropriately anonymized to create an even larger set of empirical data to improve understanding of systems engineering across the community.

5. **To date the Helix team has focused on a primarily grounded theory-based approach. The Helix team has used existing community-based documents, such as the *Systems Engineering Body of Knowledge* and the *Graduate Reference Curriculum for Systems Engineering* to provide structure to some areas of *Atlas*. Likewise, the team has begun understanding how additional research efforts relate to and could inform *Atlas*.**

   Going forward, the Helix team will examine the potential for collaboration with additional SERC research tasks as well as community based activities such as the development of the INCOSE Competency Framework and will begin folding in structures for additional research or related frameworks as appropriate, particularly existing research on systems engineering capabilities at the organizational level.

6. **The Helix team has helped several organizations begin implementation of *Atlas* in a variety of ways. However, the end goal is for *Atlas* to be implemented more widely than can be done only by direct involvement with the Helix team.**

   In order to do this, additional infrastructure must be put in place to enable widespread use by individuals or organizations, including publicly available tooling to support analysis and data collection.

With the identified gaps, the researchers noticed that Research Questions 1 (What are the characteristics of effective systems engineers?) and 2 (What makes systems engineers effective and why?) investigated have been largely addressed. Research Question 3 (What can organizations do to improve the effectiveness of its systems engineers?) has only been partially addressed. In looking to the future, the Helix team expects to shift the focus slightly, as outlined above. The primary research questions to guide the research going forward will be:

- How can an organization improve the effectiveness of its systems engineering workforce?
- How does the effectiveness of the systems engineering workforce impact the overall systems engineering capability of an organization?
- What critical factors, in additional to workforce effectiveness, are required to enable systems engineering capability?

Through looking more thoroughly at the organization, some of the research gaps will be closed, and Helix will identify workforce development patterns and qualitative findings, rather than statistically significant results. Likewise, insights into the systems engineering capabilities of organizations will be explored and modeled.

REFERENCES

"Mentor." 2014. Merriam-Webster.com. Accessed 16 April 2014. http://www.merriam-webster.com/dictionary/mentor.

"Theory." 2014. Merriam-Webster.com. Accessed 14 November 2014. http://www.merriam-webster.com/dictionary/mentor.

Bellinger, G. 2004. "Modeling and Simulation: An Introduction." Webpage. Accessed 30 August 2014. Available at: http://www.systems-thinking.org/modsim/modsim.htm

BKCASE Editorial Board. 2015. *The Guide to the Systems Engineering Body of Knowledge (SEBoK), v. 1.4*. R.D. Adcock (EIC). Hoboken, NJ: The Trustees of the Stevens Institute of Technology. Accessed 14 November 2014. www.sebokwiki.org.

Boehm, B., J. Bayuk, A. Deshmukh, R. Graybill, J.A. Lane, A. Levin, A. Madni, M. McGrath, A. Pyster, S. Tarchalski, R. Turner, and J. Wade. 2010. *Systems 2020: Strategic Initiative*. Hoboken, NJ: Systems Engineering Research Center (SERC). SERC-2010-TR-009.

Davidz, H.L. 2006. *Enabling Systems thinking to Accelerate the Development of Senior Systems Engineers*. Dissertation, Doctor of Philosophy in Engineering Systems. Massachusetts Institute of Technology.

Davidz, H.L. and D.J. Nightingale. 2007. "Enabling Systems Thinking to Accelerate the Development of Senior Systems Engineers." *Systems Engineering*. 11(1).

DoD. 2013. "SPRDE Functional Career Field: Critical Acquisition Workforce Data FY 2013-Q3 (as of June 20, 2013)." U.S. Department of Defense, Office of the Undersecretary of Defense for Acquisition, Technology, and Logistics (OUSD (AT&L)). 30 June 2013.

DoD. 2013. *Interim DoD Instruction 5000.2: Operation of the Defense Acquisition System*. Arlington, VA: U.S. Department of Defense. 26 November 2013.

Felder, W., S. Yang, M. Pennotti, K.A. Duliba, and C.Y. Mo. 2016. *RT-149: Leadership Development Framework for the Technical Acquisition Workforce*. Hoboken, NJ: Systems Engineering Research Center, Stevens Institute of Technology. SERC-2016-TR-111.

Frank, M. 2006. "Excellence in Systems Engineering: What Characterizes Successful Systems Engineers?" (Paper #106). Conference on Systems Engineering Research (CSER), April 7-8, 2006, Los Angeles, CA. 1-10.

Frank, M., K. Frampton, and T.D. Carlo. 2007. "Characteristics of Successful Systems Engineers, Systems Architects and IT Architects." Asia-Pacific Software Engineering Conference (APSEC), Nagoya, Japan. 1-12.

GAO. 2008. *Defense Acquisitions: Assessments of Selected Weapon Programs*. Washington, D.C.: Government Accountability Office (GAO). GAO-08-467SP.

GAO. 2011. *Weapons Acquisition Reform: Actions Needed to Address Systems Engineering and Developmental Testing Challenges*. Washington, D.C.: Government Accountability Office (GAO). GAO-11-806.

GAO. 2012. *Weapons Acquisition Reform: Reform Act Is Helping DOD Acquisition Programs Reduce Risk, but Implementation Challenges Remain*. Washington, D.C.: Government Accountability Office (GAO). GAO-13-103.

GAO. 2013. *High Risk Series: An Update*. Washington, D.C.: Government Accountability Office (GAO). GAO-13-283.

Goulding, C. 2002. *Grounded Theory: A Practical Guide for Management, Business and Market Researchers*. Thousand Oaks, CA: Sage.

Hutchison, N. *A Framework to Classify Experiences and Enable Career Path Analysis to Support Maturation of Effective Systems Engineers in the Defense Industry*. PhD Dissertation. Hoboken, NJ: Stevens Institute of Technology. October 2015.

Hutchison, N., A Pyster, D. Henry. 2015. “Trends in the Education of Systems Engineers.” Submitted to *Systems Engineering* March 2015.

Hutchison, N., A. Pyster, D. Henry, C. Barboza. 2015. “Understanding What Makes Systems Engineers Effective.” Submitted to *Systems Engineering* March 2015.

INCOSE Technical Operations. 2007. *Systems Engineering Vision 2020*. San Diego, CA, USA: International Council on Systems Engineering. INCOSE-TP-2004-004-02.

INCOSE. 2010. *Systems Engineering Competencies Framework*. San Diego, CA, USA: International Council on Systems Engineering. INCOSE-TP-2010-003.

INCOSE. 2012. *Systems Engineering Handbook: A Guide for System Lifecycle Processes and Activities*, version 3.2.2. San Diego, CA, USA: International Council on Systems Engineering (INCOSE), INCOSE-TP-2003-002-03.2.2.

INCOSE. 2014. *A World in Motion: Systems Engineering Vision 2025*. San Diego, CA, USA: International Council on Systems Engineering.

INCOSE. 2015. “Certification.” Website of the International Council on Systems Engineering (INCOSE). Accessed April 29, 2015. Available at www.incose.org/certification.

ISO/IEC. 2009. "Systems and Software Engineering Vocabulary (SEVocab)" - ISO/IEC 24765. in International Organization for Standardization (ISO)/International Electrotechnical Commission (IEC) [database online]. Geneva, Switzerland, 2009 [cited December 21 2009]. Available at: http://pascal.computer.org/sev_display/index.action.

ISO/IEC/IEEE. 2008. *Systems and Software Engineering -- System Lifecycle Processes.* Geneva, Switzerland: International Organisation for Standardisation/International Electrotechnical Commissions. ISO/IEC/IEEE 15288:2008.

ISO/IEC/IEEE. 2010. *Systems and Software Engineering - System and Software Engineering Vocabulary (SEVocab)*. Geneva, Switzerland: International Organization for Standardization (ISO)/International Electrotechnical Commission (IEC)/ Institute of Electrical and Electronics Engineers (IEEE). ISO/IEC/IEEE 24765:2010.

PMI. 2012. *PMI Lexicon of Project Management Terms*, Version 2.0. Newton Square, PA: Project Management Institute (PMI).

Pyster, A. (ed.). 2009. *Graduate Software Engineering 2009 (GSwE2009): Curriculum Guidelines for Graduate Degree Programs in Software Engineering*. Integrated Software & Systems Engineering Curriculum Project. Hoboken, NJ, USA: Stevens Institute of Technology, September 30, 2009.

Pyster, A., D.H. Olwell, T.L.J. Ferris, N. Hutchison, S. Enck, J. Anthony, and D. Henry (eds.). 2015. *Graduate Reference Curriculum for Systems Engineering (GRCSE®) v.1.1*. Hoboken, NJ, USA: Trustees of the Stevens Institute of Technology.

Pyster, A., P. Dominick, D. Henry, N. Hutchison, C. Lipizzi, M. Kamil, S. Manchanda. 2014(b). Atlas*: The Theory of Effective Systems Engineers*, v. 0.25. Hoboken, NJ: Systems Engineering Research Center, Stevens Institute of Technology. SERC-2014-TR-038-4.

Pyster, A., R.L. Pineda, D. Henry, N. Hutchison. 2013. *Helix – Phases 1 and 2.* Hoboken, NJ: Systems Engineering Research Center, Stevens Institute of Technology. SERC-2013-TR-038-2.

Pyster, A., R.L. Pineda, D. Henry, N. Hutchison. 2014. *Helix – Phase 3*. Hoboken, NJ: Systems Engineering Research Center, Stevens Institute of Technology. SERC-2014-TR-038-3.

SEI. 2007. *Capability maturity model integrated (CMMI) for development, version 1.2*. Pittsburg, PA, USA: Software Engineering Institute (SEI)/Carnegie Mellon University (CMU).

Sheard, S. 1996. "Twelve Systems Engineering Roles." Proceedings of the International Council on Systems Engineering (INCOSE) Sixth Annual International Symposium, Boston, Massachusetts, USA, 1996.

Sheard, S. 2000. "Systems Engineering Roles Revisited." Software Productivity Consortium, NFP.

SHRM. 2015. "HR Terms: Career Path." Alexandria, Virginia: Society for Human Research Management website. Last updated 2015. Accessed May 27, 2015. Available at: http://www.shrm.org/templatestools/glossaries/hrterms/pages/c.aspx.

Torelli, N. 2012. Presentation as part of Pyster, A.; Torelli, N.; Madni, A.; Hanneman, L.; Highland, F. 2012. "Systems Engineering Degrees and Certificates: What is the Right Mix for Industry and Government?" *National Defense Industrial Association (NDIA) 15th Annual Systems Engineering Conference*, October 22-25, 2012, San Diego, CA.

Welby, S. 2010. "DoD Systems Engineering Update." National Defense Industrial Association (NDIA) Gulf Coast Chapter: 36th Air Armament Symposium, Fort Walton Beach, FL, USA. 1-25.

Welby, S. 2011. "DoD Systems Engineering: Status Report." National Defense Industrial Association (NDIA) Systems Engineering (SE) Division Meeting, Washington D.C. Metropolitan Area, USA. 1-33.

GLOSSARY

ACRONYMS AND ABBREVIATIONS

CSE	Chief Systems Engineer
DASD(SE)	U.S. Deputy Assistant Secretary of Defense for Systems Engineering
DIB	Defense Industrial Base (that supports DoD)
DoD	U.S. Department of Defense
GRCSE	Graduate Reference Curriculum for Systems Engineering
HR	Human Resources
INCOSE	International Council on Systems Engineering
IPT	Integrated Product Team
IR&D	Internal (or Independent) Research & Development
IRB	Internal Review Board
IT	Information Technology
IV&V	Integration, (or Independent) Verification, & Validation
MBA	Master of Business Administration
NDIA-SED	National Defense Industrial Association – Systems Engineering Division
PEO	Program Executive Office
PLM	Product Life Management
QRC	Quick Reaction Capability
SE	Systems Engineering
SERC	Systems Engineering Research Center
SEBoK	*Guide to the Systems Engineering Body of Knowledge*
SME	Subject Matter Expert
SPRDE	Systems Planning, Research, Development, and Engineering
UARC	University-Affiliated Research Center
V&V	Verification & Validation
PM	Project (or Program) Manager

GLOSSARY

Consistency in the definition and understanding of terminology and concepts is essential for any deliberation. This section presents the definitions and classifications that are relevant to *Atlas*. Some have been obtained from available literature, while others have been created specifically for *Atlas*.

- **Systems Engineer**

> *A* ***Systems Engineer*** *is an individual who performs systems engineering activities and is recognized (either formally or informally) by his or her organization for her ability to perform these activities.*

This definition of a systems engineer does *not* refer to the title that someone may hold in her organization. Someone may never hold the title ‘Systems Engineer’, but could be considered to be one based on the activities she performs. Similarly, someone may hold the title ‘Systems Engineer’, but her activities may not be considered to be systems engineering activities.

- **Effective Systems Engineer**

> *An **Effective Systems Engineer** is someone who consistently delivers value by performing systems engineering activities in positions assigned by the organization.*

This definition is fundamental to *Atlas* since the focus of Helix research is the effectiveness of systems engineers. Though 'effectiveness' is a subjective term, this definition ties it to 'value' that can be defined and even measured – qualitatively, if not quantitatively.

- **Value**

> ***Value** describes the benefits gained through the application of systems engineering activities, as distinct from benefits gained through other disciplines.*

As an effective systems engineer is one who consistently delivers value, the definition of value and the values expected by an organization are a critical input to effectiveness. Section 4.2 provides a description of the common values seen across the Helix sample.

- **Chief Systems Engineer (CSE)**

> *A **Chief Systems Engineer (CSE)** is one who has formal responsibility to oversee and shepherd the technical correctness and to maintain a consistent vision for a system, often coordinating with many other systems engineers who have smaller scopes of responsibility.*

The Chief Systems Engineer (CSE) position is one of the most senior technical positions that system engineers can achieve while staying in a technical track (as opposed to a management track). Though the title 'Chief Systems Engineer' is not used in all organizations, the concept of a CSE position (or equivalent) is common, especially in industry. There is no consistent description of a CSE's (or equivalent's) formal authority, but overall responsibility for a system is often split in some way between the CSE and the project or program manager (PM).

- **Position**

> *A **Position** is the particular arrangement of roles and responsibilities for an individual, as defined and assigned by the organization.*

Often, positions are equivalent to an individual's title. Using this definition, a position is usually specific to an organization and a specific position does not translate across organizations.

- **Role**

> *A **Role** performed by an individual consists of a specific set of related systems engineering activities.*

Typically, an individual performs multiple roles in any given position. Unlike positions, roles are consistent sets of activities that are applicable across organizations, domains, etc.

- **Career Path**

  > *An individual's* ***Career Path*** *is the precise combination (in terms of characteristics, timing, and order) of experiences, mentoring, and education and training that they undergo during their entire career.*

  This definition, created for *Atlas*, is different from how career paths are typically defined in the human resources (HR) community. HR definitions tend to be focused on rigid hierarchy that may be useful for HR classification and management of positions within an organization. However, they provide little insight into the growth and development of individuals throughout their career, particularly across organizations.

- **Proficiency**

  > *The* ***Proficiency*** *of an individual is the quality or state of knowledge, skills, abilities, behaviors, and cognition.*

  In *Atlas*, the term 'proficiency' is used broadly to include everything that an individual needs to be good at in order to be an effective systems engineer. This distinguishes *Atlas* from competency models that tend to focus primarily on the discipline of systems engineering.

APPENDIX A: HELIX PUBLICATIONS

Journal Article

Hutchison, N., A. Pyster, D. Henry. "Atlas: Understanding What Makes Systems Engineers Effective in the US Defense Community." *Systems Engineering. In process* - expected to be published in January 2017.

Conference Papers and Presentations

Henry, D., Hutchison, N., Pyster, A., Dominick, P., Lipizzi, C., Kamil, M., Manchanda, S. “Summary of Findings from the Helix Project (2013-14) – An Investigation of the DNA of the Systems Engineering Workforce,” National Defense Industries Association Systems Engineering Conference, Springfield, VA, October 27-30, 2014

Henry, D., N. Hutchison, A. Pyster, P. Dominick, C. Lipizzi, M. Kamil, S. Manchanda. 2014. “Summary of Findings from the Helix Project (2013-14) - An Investigation of the DNA of the Systems Engineering Workforce”. Proceedings of the National Defense Industrial Association (NDIA) Systems Engineering Conference, Springfield, VA, USA, October 28-30, 2014.

Henry, D., N. Hutchison, A. Pyster. 2015. “Summary of Findings from the Helix Project (2013-2014): An Investigation of the DNA of the Systems Engineering Workforce.” Proceedings of the National Defense Industrial Association (NDIA) Systems Engineering Conference, October 29-November 1, 2014, Crystal City, VA.

Hutchison, N. *A Framework to Classify Experiences and Enable Career Path Analysis to Support Maturation of Effective Systems Engineers in the Defense Industry*. PhD Dissertation. Hoboken, NJ: Stevens Institute of Technology. October 2015.

Hutchison, N. and A. Pyster. 2015. “The Helix Project: Analysis of INCOSE SE Certification Program Applicants.” Presentation to the INCOSE Corporate Advisory Board, INCOSE International Workshop, January 2015, Los Angeles, CA.

Hutchison, N., D. Henry, A. Pyster, and P. Dominick. 2014. “Systems Engineering Career Analysis: Supporting a Theory of Systems Engineers.” Proceedings of the INCOSE Europe, Middle-East, Africa Systems Engineering Conference (EMEASEC), Capetown, South Africa, October 27-30, 2014.

Hutchison, N., D. Henry, A. Pyster, P. Dominick. 2014(b). “Systems Engineering Career Analysis: Supporting a Theory of Systems Engineers.” Proceedings of the Europe, Middle East, and Asia Sector Systems Engineering Conference, 27-30 October 2014, Cape Town, South Africa.

Hutchison, N., D. Henry, A. Pyster. 2014(a). “Early Findings from Interviewing Systems Engineers Who Support the US Department of Defense.” Proceedings of the International Council on Systems Engineering (INCOSE) International Symposium, 30 June – 4 July 2014, Las Vegas, NV.

Hutchison, N., Henry, D., Pyster, A., Pineda, R., “Early Findings from Interviewing Systems Engineers who Support the US Department of Defense”, 2014 International Council on Systems Engineering International Symposium, June 30-July 3, 2014, Las Vegas, Nevada

Jauregui, C., A. Pyster, D. Henry, N. Hutchison, and C. Wright. 2016. “Insights on the Experiences and Education of INCOSE-Certified Expert Systems Engineering Professionals and Chief Systems Engineers.” To be published in the Proceedings of the International Council on Systems Engineering (INCOSE) 26th International Symposium, 18-21 July 2016, Edinburgh, Scotland.

Lipizzi, C., S. Manchanda, M. Kamil, A. Pyster, D. Henry, N. Hutchison. 2015. “The Education Background of INCOSE Systems Engineering Professional Certification Program Applicants.” Proceedings of the International Council on Systems Engineering (INCOSE) International Symposium, Seattle, WA, July 2015. (Accepted for publication)

Pyster, A. 2015. "The Helix Project: Analysis of INCOSE SE Certification Program Applicants." Special Presentation to the International Council on Systems Engineering (INCOSE) Corporate Advisory Board (CAB), January 25, 2015 as part of the INCOSE International Workshop, January 25-27, Los Angeles, CA.

Pyster, A., D. Henry, N. Hutchison, C. Jauregui, J. Armstrong, M. Clifford. 2015. Report on the Second Helix Workshop: Exploring the Theory of Systems Engineers' Effectiveness. Hoboken, NJ: Systems Engineering Research Center, Stevens Institute of Technology.

Pyster, A., N. Hutchison, D. Henry, C. Barboza. 2015. "Helix: What Makes Systems Engineers Effective?" Proceedings of the Conference on Systems Engineering Research (CSER), March 17-19, 2015, Hoboken, NJ.

Pyster, A., P. Dominick, D. Henry, N. Hutchison, C. Lipizzi, M. Kamil, S. Manchanda. 2014(b). Atlas*: The Theory of Effective Systems Engineers*, v. 0.25. Hoboken, NJ: Systems Engineering Research Center, Stevens Institute of Technology. SERC-2014-TR-038-4.

Pyster, A., P. Dominick, D. Henry, N. Hutchison, C. Lipizzi, M. Kamil. 2014(a). Report on the First Helix Workshop: Exploring the Theory of Systems Engineers' Effectiveness. Hoboken, NJ: Systems Engineering Research Center, Stevens Institute of Technology.

Pyster, A., R. Pineda, D. Henry, N. Hutchison. "Helix: Investigating the DNA of the Systems Engineering Workforce." Proceedings of the National Defense Industrial Association (NDIA) Systems Engineering Conference, October 28-31, 2013, Crystal City, VA.

Pyster, A., R.L. Pineda, D. Henry, N. Hutchison. 2013. *Helix – Phases 1 and 2.* Hoboken, NJ: Systems Engineering Research Center, Stevens Institute of Technology. SERC-2013-TR-038-2.

Pyster, A., R.L. Pineda, D. Henry, N. Hutchison. 2014. *Helix – Phase 3*. Hoboken, NJ: Systems Engineering Research Center, Stevens Institute of Technology. SERC-2014-TR-038-3.

Pyster, A., Rifkin, S., Henry, D., Lasfer, K., Hutchison, N., Gelosh, D., "The Helix Project: Understanding the Systems Engineers Who Support the US Department of Defense", 2013 Asia Pacific Conference on Systems Engineering, September 8-11, 2013, Kyoto, Japan

Squires, A., Wade, J., Hutchison, N. 2016. "Building a Pathway to Systems Education for the Global Engineer." To be published in the Proceedings of the American Society for Engineering Education (ASEE) 123rd Annual Conference, 26-29 June 2016, New Orleans, Louisiana, USA.

Pyser, A., S. Rifkin, D. Henry, K. Lasfser, N. Hutchison, D. Gelosh. 2013. "The Helix Project: Understanding the Systems Engineers who Support the US Department of Defense." Asia-Pacific Conference on Systems Engineering (APCOSE), Yokahama, Japan, Sept. 8 - 11, 2013.

In process

A presentation on *Atlas*, "Atlas: A Guide to Growing Effective Systems Engineers," has been accepted for the 2017 Institute of Industrial and Systems Engineers (IISE) to be held in Orlando, FL. Nicole Hutchison is the presenter.

A paper on the roles presented in *Atlas*, "The Roles of Systems Engineers Revised," has been submitted for the 2017 INCOSE International Symposium. The authors are Nicole Hutchison and Jon Wade.

Other

ABET Symposium 2016, Fort Lauderdale, FL – ABET panel on systems engineering education and research for the 2016 ABET conference. Nicole Hutchison presented on Helix.

INCOSE Healthcare Systems Engineering Working Group Webinar – November 29, 2016. Nicole Hutchison delivered a webinar, a 60-minute overview of Atlas with specific implications related to healthcare systems engineers.

APPENDIX B: SELF-ASSESSMENT TOOLS (PAPER BASED)

This appendix provides the paper-based tools for assessment generated by the Helix team. These tools are easy and simple for an individual to use to gain insight into his or her career or for an organization to deploy to enable career planning with its employees. The materials include the templates and some basic guidance on how to use them. However, these materials do not include the depth of detail included in the Excel-based templates. Particularly if an organization intends to collect data from a high percentage of its employees based on *Atlas*, it is recommended that the Excel-based files be used. They are available at http://www.sercuarc.org/projects/helix/ under "Deliverables."

The content of the tools is outlined below:

INSTRUCTIONS FOR COMPLETING A PROFICIENCY SELF-ASSESSMENT

Overview

Proficiency defines the knowledge, skills, abilities, behaviors, patterns of thinking, and abilities that are critical to the effectiveness of systems engineers. The *Atlas* proficiency model consists of six difference proficiency areas:

- **Math/Science/General Engineering**: Foundational concepts from mathematics, physical sciences, and general engineering;
- **System's Domain & Operational Context**: Relevant domains, disciplines, and technologies for a given system and its operation;
- **Systems Engineering Discipline**: Foundation of systems science and systems engineering knowledge;
- **Systems Engineering Mindset**: Skills, behaviors, and cognition associated with being a systems engineer;
- **Interpersonal Skills**: Skills and behaviors associated with the ability to work effectively in a team environment and to coordinate across the problem domain and solution domain; and
- **Technical Leadership**: Skills and behaviors associated with the ability to guide a diverse team of experts toward a specific technical goal.

Each of these areas contains several categories, or groupings of related knowledge, skills, abilities, behaviors, or cognitions, as illustrated in Table 1.

Self-Assessment

In order to perform a self-assessment, individuals are asked to review the definitions of the proficiency areas above and the categories in Table 1. Additional detail can be found in the full report on *Atlas 1.0*, SERC-2016-TR-118, found at the Helix webpage (http://www.sercuarc.org/projects/helix/). Then use the template to generate a "0 to 10" initial assessment of your current proficiency in each Area, with "0" meaning you have no skill in the area and 10 meaning your skills are the top within your experiences. Consider the following guidelines:

- For each Proficiency Area, think about proficiency across *all categories*, not just one. For example, if you are a "10" in a single category, but a "5" in all others, you would not be a "10" for the entire Area.
- For each Area, think about what is most critical for your current position. This may not change your assessment, but may mean that a lower number not an issue.
- Consider your past experiences in the Area, any training or education that might be relevant, and where you might have received guidance from a mentor or leader. These things together will have shaped your proficiency, and thinking about them may help you to assess yourself more realistically.
- You know your strengths and areas for growth – be honest in your responses.

A proficiency rubric for further guidance can be found on page 78.

Once you have completed your initial assessment for your *current* proficiency, you can choose to retroactively assess what your proficiency was at different points in your career. For example, when you completed your undergraduate education or joined your current organization. This may help you to better reflect on changes over time. If you do this, revisit your current proficiency assessment afterwards and determine whether any adjustments are required.

Area	Category	Topic
1. Math / Science / General Engineering	1.1. Natural Science Foundations	
	1.2. Engineering Fundamentals	
	1.3. Probability and Statistics	
	1.4. Calculus and Analytical Geometry	
	1.5. Computing Fundamentals	
2. Systems' Domain & Operational Context	2.1. Principal and Relevant Systems	< List of Principal and Relevant Systems >
	2.2. Familiarity with Principal System's Concept of Operations (ConOps)	
	2.3. Relevant Domains	< List of relevant Domains >
	2.4. Relevant Technologies	< List of relevant Technologies >
	2.5. Relevant Disciplines and Specialties	< List of relevant Disciplines and Specialties >
	2.6. System Characteristics	< List of applicable System Types, Scales, and Levels >
3. Systems Engineering Discipline	3.1. Lifecycle	3.1.1 Lifecycle Models 3.1.2 Concept Definition 3.1.3 System Definition 3.1.4 System Realization 3.1.5 System Deployment and Use 3.1.6 Product and Service Life Management
	3.2. Systems Engineering Management	3.2.1 Planning 3.2.2 Risk Management 3.2.3 Configuration Management 3.2.4 Assessment and Control 3.2.5 Quality Management
	3.3. SE Methods, Processes, and Tools	3.3.1 Balance and Optimization 3.3.2 Modeling and Simulation 3.3.3 Development Process 3.3.4 Systems Engineering Tools
	3.4. Systems Engineering Trends	3.4.1 Complexity 3.4.2 Model Oriented Systems Engineering 3.4.3 Systems Engineering Analytics 3.4.4 Agile Systems Engineering
4. Systems Engineering Mindset	4.1. Big-Picture Thinking	
	4.2. Paradoxical Mindset	4.2.1 Big-Picture Thinking and Attention to Detail 4.2.2 Strategic and Tactical 4.2.3 Analytic and Synthetic 4.2.4 Courageous and Humble 4.2.5 Methodical and Creative
	4.3. Flexible Comfort Zone	
	4.4. Abstraction	
	4.5. Foresight and Vision	
5. Interpersonal Skills	5.1. Communication	5.1.1 Audience 5.1.2 Content 5.1.3 Mode
	5.2. Listening and Comprehension	
	5.3. Working in a Team	
	5.4. Influence, Persuasion and Negotiation	
	5.5. Building a Social Network	
6. Technical	6.1. Building and Orchestrating a Diverse	

Area	Category	Topic
Leadership	Team	
	6.2. Balanced Decision Making & Rational Risk Taking	
	6.3. Guiding Stakeholders with Diverse/Conflicting Needs	
	6.4. Conflict Resolution & Barrier Breaking	
	6.5. Business and Project Management Skills	
	6.6. Establishing Technical Strategies	
	6.7. Enabling Broad Portfolio-Level Outcomes	

PROFICIENCY SELF-ASSESSMENT RUBRIC

Atlas Proficiency Area / Category	Proficiency Level "1"	Proficiency Level "3"	Proficiency Level "5"
1. Math / Science / General Engineering			
1.1. Natural Science Foundations	Minimal awareness of the basic concepts of physics, chemistry, and biology		Expert in the principles and concepts of physics, chemistry and biology including practical experience, and ability to apply these in the system's context
1.2. Engineering Fundamentals	Minimal awareness of the basic engineering concepts, processes, and techniques.		Expert in basic engineering concepts, processes, and techniques including practical experience, and ability to apply these in the system's context
1.3. Probability & Statistics	Minimal awareness of the basics of probability and statistics		Expert in probability theory, probability distributions, statistical measures and other related topics, and ability to readily apply them where required
1.4. Calculus & Analytical Geometry	Minimal awareness of differential calculus, integral calculus, coordinate systems, and geometric equations		Expert in differential and integral calculus methods, coordinate systems, transformations, describing and analyzing geometric objects and ability to readily apply them where required
1.5. Computing Fundamentals	Minimal awareness of computer organization, operating systems, and programming languages		Expert in computer architectures, networking, operating systems, programming languages and ability to readily apply them where required
2. Systems' Domain & Operational Context			
2.1. Principal and Relevant Systems	Minimal knowledge about the specific systems		Expert in the specific systems, their development and operation
2.2. Familiarity with System's Concept of Operations (ConOps)	Minimal awareness of the ConOps of the principal system		Expert in the ConOps of the system, and the ability to comprehensively develop ConOps
2.3. Relevant Domains	Minimal familiarity with the terminology and basic concepts of the specific domains		Expert in the domain and the development and operation of systems in that domain.
2.4. Relevant Technologies	Minimal familiarity with the terminology and basic concepts of the specific technologies		Expert in the technology and its current development, and the ability to easily apply it to system development

Atlas Proficiency Area / Category	Proficiency Level "1"	Proficiency Level "3"	Proficiency Level "5"
2.5. Relevant Disciplines and Specialties	Minimal familiarity with the terminology and basic concepts of the specific disciplines		Expert in the discipline and latest advancements
2.6. System Characteristics	Minimal familiarity with the specific Types, Scales, and Levels of systems		Expert in the specific Types, Scales, and Levels of systems
3. Systems Engineering Discipline			
3.1. Lifecycle	Minimal awareness of lifecycle models and lifecycle stages		Expert in the understanding of lifecycle models and how systems are developed in them. A deep understanding of specific lifecycle stages of system development and ability to carry out the required technical activities at those stages
3.2. Systems Engineering Management	Minimal awareness of systems engineering management activities		Expert in specific topics of systems engineering management and ability to perform the required management activities
3.3. SE Methods, Processes, & Tools	Minimal awareness of SE methods, processes and tool in an isolated manner		Expert in specific SE methods, processes, and tools, and in the application of these.
3.4. Systems Engineering trends	Minimal awareness of the specific trends and their application to systems development		Expert in the specific trends and their application to systems development
4. Systems Engineering Mindset			
4.1. Big-Picture Thinking	Minimal ability to think beyond a narrow scope of the problem at hand	Able to think in a limited manner outside a narrow scope with some guidance	Expert in thinking broadly along various dimensions (e.g., regarding broader domain or enterprise-level considerations, and linking across apparent disparate domains such as incorporating "soft" science with "hard" science)
4.2. Paradoxical Mindset	Minimal ability to handle seemingly opposed views	Able to understand the one of the opposed views separately but not both	Expert in the understanding of two opposed views and perspectives, ability to successfully handle them both separately and together, and the ability to successfully move from one perspective to another

Atlas Proficiency Area / Category	Proficiency Level "1"	Proficiency Level "3"	Proficiency Level "5"
4.3. Flexible Comfort Zone	Comfortable only strictly within one's comfort zone and area of technical expertise	Able to permeate beyond one's comfort zone in a limited manner, but hesitates to explore the unknown	Willing and able to permeate the boundaries of one's comfort zone with ease, and able to comfortably explore the unknown and readily seek interdisciplinary SME
4.4. Multi Scale Abstraction	Minimal ability to abstract or infer from individual pieces of information and relate to environmental context	Able to abstract insights with some guidance and prior experience and understand system in larger operational context	Expert in quickly and effectively abstracting (from highly detailed level to highly conceptual level) new and significant insights from seemingly disparate pieces of information across system and environmental scales
4.5. Foresight & Vision	Minimal ability to comprehend future impacts of current decisions and situations	Able to comprehend impacts in the near future, in a limited manner	Expert in seeing future impacts of current decisions, and to clearly visualize future stages of a system's lifecycle
5. Interpersonal Skills			
5.1. Communication	Minimal ability to successfully communicate any information to any audience in any mode	Able to communicate well in one predominant mode with limited familiar audience	Expert in being able to successfully and unambiguously communicate to a variety of audience and a wide range of technical and non-technical content, in various written and oral modes.
5.2. Listening & Comprehension	Minimal ability to listen to and understand others' points and perspectives	Able to listen to other's points, but limited ability to comprehend	Expert in listening and successfully comprehending others' points and perspectives
5.3. Working in a Team	Minimal ability to work with anyone else, preferring to work alone	Able to work in familiar teams, but limited ability to work on new teams	Very comfortable to work with others, and being able to quickly and successfully become part of any team exhibiting positive team dynamics
5.4. Influence, Persuasion & Negotiation	Minimal ability to modify another person's viewpoint or perspective, even when that is detrimental	Able to influence others in a limited manner, only with familiar individuals or when they are not experts in their own fields	Expert in positively influencing others, particularly experts in their own fields, to see beyond their viewpoints and to come to agreements for the good of the overall system

Atlas Proficiency Area / Category	Proficiency Level "1"	Proficiency Level "3"	Proficiency Level "5"
5.5. Building a Social Network	Minimal ability to form any social relationship with a professional acquaintance	Able to form a limited social network among those with frequent interactions	Expert in establishing strong social relationships with professional acquaintances both within and outside the organization
6. Technical Leadership			
6.1. Building & Orchestrating a Diverse Team	Minimal ability to form or lead a team with any success	Able to build a team with guidance but has difficulty in handling or delegating to a diverse team	Expert in bringing together the right team for the task, being able to synergistically draw individual strengths of team members, successfully leading the team to achieve end goal
6.2. Balanced Decision Making & Rational Risk Taking	Minimal ability to take balanced decisions or to take any rational risks	Able to take some balanced decisions with some guidance, but limited ability to take rational risks	Expert in taking successful decision considering all relevant factors and constraints, and being able to rationally calculate risks when required
6.3. Guiding Stakeholders with Diverse/ Conflicting Needs	Minimal ability to guide internal and external stakeholders and their needs	Able to guide familiar stakeholders, who have well established needs, in a limited manner	Expert in leveraging good relationships with internal and external stakeholder, and successfully meeting their needs
6.4. Conflict Resolution & Barrier Breaking	Minimal ability to resolve any conflict that negatively affects the system, and unable to break barriers of opinions and perspectives that prevent any progress	Able to resolve minor conflict mostly among familiar individuals	Expert in successfully resolving conflict between individuals or teams for the sake of the overall system, and able to break down various technical or cultural barriers
6.5. Business & Project Management Skills	Minimal ability to perform business and project management activities	Able to perform business and project management activities with some guidance and reference	Expert in the knowledge, understanding, and application of various business and project management skills.
6.6. Establishing Technical Strategies	Tactical approach to technology on a project-by-project basis		Develops technical strategies that impact multiple projects (e.g. investment decisions, prioritization of technology roadmaps, etc.)

Atlas Proficiency Area / Category	Proficiency Level “1”	Proficiency Level “3”	Proficiency Level “5”
6.7. Enabling Broad Portfolio-Level Outcomes	Focuses only on outcomes for individual projects		Identifies issues and opportunities that impact an entire portfolio of systems Communicates these issues to leadership and engineers

Proficiency Self-Assessment Tool

Most critical category:
Least critical category:

Start of Career:

Today:

Target State:

Most critical category:
Least critical category:

Start of Career:

Today:

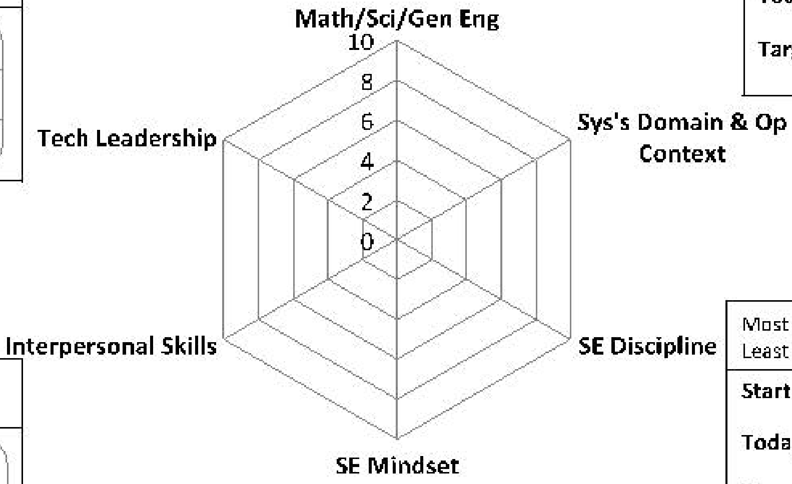
Target State:

Most critical category:
Least critical category:

Start of Career:

Today:

Target State:



Most critical category:
Least critical category:

Start of Career:

Today:

Target State:

Most critical category:
Least critical category:

Start of Career:

Today:

Target State:

Most critical category:
Least critical category:

Start of Career:

Today:

Target State:

Current Position: ______________
Today's Date: ______________
Start of Career Date: ______________
Target Position: ______________
Target By Date: ______________

INSTRUCTIONS FOR COMPLETING A CAREER PATH ASSESSMENT

Overview

An individual's career path is the precise combination of experiences, mentoring, education, and training that an individual goes, particularly their characteristics, timing, and order. In order to complete a career assessment, an individual should work through the steps outlined here while filling out the career path template.

Experiences

The Helix team chose to use a **position** as the unit of measure for experiences; a position is established by the organization and defines the roles and responsibilities to be performed.

Based on both the literature and the Helix data itself, each position has several characteristics:

- **Relevance:** A 'relevant' position is one that enables a systems engineer to develop the proficiencies critical to systems engineering. Determine a starting point for relevant experiences; this will become the first position (P1) of the career path. Fill in the title and the year(s) for the position(s).
- **Organizations:** Fill out the name of the organization for each position. This will help to show any transition or variation between organizations.
- **Roles:** A role is a collection of related systems engineering activities. Roles were identified based on the activities consistently performed by systems engineers. There are 16 roles identified in *Atlas*, as described in Table 1, below. For each position, review your activities and responsibilities and write down *all* roles played during that position.
- **Lifecycle Phases:** Generic systems engineering lifecycle phases considered in *Atlas* are based on the lifecycle phases in the *Guide to the Systems Engineering Body of Knowledge (SEBoK*), as explained on page 5. (BKCASE Authors 2015) For each position, fill in the area(s) of the lifecycle you worked on.
- **Key Milestones.** Note any key changes in types of positions under key milestones. For example, first systems engineering role, first chief systems engineer role, first supervisory position, etc. would all be indicators of change or growth over career.

Education and Training

Note any educational milestones or key training milestones with the position/timeline in which they occurred. Education milestones may include the completion of a degree or participation in a course that was particularly relevant or impactful for your career. Key training is training that was particularly impactful or useful for your career. You do not need to include training that did not have an impact.

Other

Your organization may ask you to add other information, such as participation in professional societies, publications, etc. to your career path.

Role Name	Role Description
Roles Focused on Systems	
Concept Creator	Individual who holistically explores the problem or opportunity space and develops the overarching vision for a system(s) that can address this space. A major gap pointed out to the Helix team – particularly when working to implement the findings of Helix – has been that of the development of an overarching system vision. This is a critical first step in the systems lifecycle, and several organizations stated that they believed it needed to be separated out. In addition, when looking to the future of what systems engineers need to do (e.g. INCOSE Vision 2025 (2015)), the focus on early engagement and setting the vision was deemed critical.
Requirements Owner	Individual who is responsible for translating customer requirements to system or sub-system requirements; or for developing the *functional* architecture. This is unchanged from (Sheard 1996).
System Architect	Individual who owns or is responsible for the architecture of the system. This is an update of Sheard's "System Designer" role (1996). There was concern both at community events and during later interviews that nowhere in the presented framework did the critical role of systems engineers in architecture come out clearly. Some also argued that "Design" gave the impression that this roles focuses specifically on the details of systems design over architecture.
System Integrator	Individual who provides a holistic perspective of the system; this may be the 'technical conscience' or 'seeker of issues that fall in the cracks' – particularly, someone who is concerned with interfaces. Likewise, there was concern over the word "Glue", which many expressed was not clearly descriptive enough.
System Analyst	Individual who provides modeling or analysis support to system development activities, and helps to ensure that the system as designed meets he specification. This is unchanged from Sheard's roles (1996).
Detailed Designer	Individual who provides technical designs that match the system architecture; an individual contributor in any engineering discipline who provides part of the design for the overall system. This is an addition based on the Helix data. While systems engineers do not always get involved with detailed design, in smaller organizations or on smaller projects it is more common. Likewise, systems engineers who had played this role explained that it was critical in developing their own technical and domain expertise as well as in understanding the design approaches of classic engineers.
V&V Engineer	Individual who plans, conducts, or oversees verification and validation activities such as testing, demonstration, and simulation. This is unchanged from Sheard's roles (1996).
Support Engineer	Individual who performs the 'back end' of the systems lifecycle, who may operate the system, provide support during operation, provide guidance on maintenance, or help with disposal. This was previously titled "Logistics and Operations Engineer" in Sheard (1996). However, in interviews and at community events, the Helix team received feedback that using this title gave the impression that this role was limited and did not encompass the full spectrum of systems engineers' activities at system deployment or post-deployment. Likewise, in several organizations, "logistics" and "operations" were seen as separate disciplines from systems engineering, which caused some contention in discussions. The renaming of this category is intended to address these issues.

Roles Focused on SE Process and Organization	
Systems Engineering Champion	Individual who promotes the value of systems engineering to individuals outside of the SE community - to project managers, other engineers, or management. This may happen at the strategic level or could involve looking for areas where systems activities can provide a direct or immediate benefit on existing projects. Sheard recommended that a role such as this, labeled in her work, as “Systems Engineering Champion”, be added in (2000).
Process Engineer	Individual who defines and maintains the systems engineering processes as a whole and who also likely has direct ties into the business. This individual provides critical guidance on how systems engineering should be conducted within an organization context. This is unchanged from Sheard’s roles (1996).
Roles Focused on Teams	
Customer Interface	Individual who coordinates with the customer, particularly for ensuring that the customer understands critical technical detail and that a customer’s desires are, in turn, communicated to the technical team. This is unchanged from Sheard’s roles (1996).
Technical Manager	Individual who controls cost, schedule, and resources for the *technical* aspects of a system; often someone who works in coordination with an overall project or program manager. This is unchanged from Sheard’s roles (1996).
Information Manager	Individual who is responsible for the flow of information during system development activities. This includes the systems management activities of configuration management, data management, or metrics. This is unchanged from Sheard’s roles (1996).
Coordinator	Individual who brings together and brings to agreement a broad set of individuals or groups who help to resolve systems related issues. This is a critical aspect of the management of teams. This is unchanged from Sheard’s roles (1996).
Instructor/Teacher	Individual who is provides or oversees critical instruction on the systems engineering discipline, practices, processes, etc. This can include the development or delivery of training curriculum as well as academic instruction of formal university courses related to systems engineering. While any discipline could conceivably have an instructor role, this denotes a focus on systems and is a critical component in the development of an effective systems engineering workforce. This is an addition to the Sheard roles (1996)

Systems Engineering Lifecycle

- **Concept Definition** - A set of core technical activities of SE in which the problem space and the needs of the stakeholders are closely examined. This consists of analysis of the problem space, business or mission analysis, and the definition of stakeholder needs for required services within it.
- **System Definition** - A set of core technical activities of SE, including the activities that are completed primarily in the front-end portion of the system design. This consists of the definition of system requirements, the design of one or more logical and physical architectures, and analysis and selection between possible solution options.
- **System Realization** - The activities required to build a system, integrate disparate system elements, and ensure that a system both meets the needs of stakeholders and aligns with the

requirements identified in the system definition stage. This includes integration, verification, and validation (IV&V).

- **System Deployment and Use** - A set of core technical activities of SE to ensure that the developed system is operationally acceptable and that the responsibility for the effective, efficient, and safe operations of the system is transferred to the owner. Considerations for deployment and use must be included throughout the system life cycle. Activities within this stage include deployment, operation, maintenance, and logistics.
- **Product and Service Life Management** - Deals with the overall life cycle planning and support of a system. The life of a product or service spans a considerably longer period of time than the time required to design and develop the system. This stage includes service life extension, updates, upgrades, and modernization, and disposal and retirement. The organizations in the current sample are primarily concentrated on new development, so this is a very under-represented aspect of the life cycle.
- In addition to these life cycle phases, the SEBoK includes orthogonal activities of systems engineers, **Systems Engineering Management**, defined as managing the resources and assets allocated to perform SE activities. Activities include planning, assessment and control, risk management, measurement, decision management, configuration management, information management, and quality management. These activities can occur at any point in the systems engineering lifecycle.

Career Path Self-Assessment Tool

Date: ________________

Concept Definition										
System Definition										
System Realization										
System Deployment and Use										
Product and Service Life Management										
Systems Engineering Management										
Role(s) Performed										
Domain(s)										
System Characteristics										
Position										
Organization(s)										
Dates										
Milestones (Key positions, education, or training)										