ARE WE TEACHING SYSTEMS ENGINEERING STUDENTS WHAT THEY NEED TO KNOW?

by

Tracy El Khoury

A Thesis

Submitted to the Faculty of Purdue University

In Partial Fulfillment of the Requirements for the degree of

Master of Science in Aeronautics and Astronautics

School of Aeronautics and Astronautics

West Lafayette, Indiana

August 2020

THE PURDUE UNIVERSITY GRADUATE SCHOOL
STATEMENT OF COMMITTEE APPROVAL

Dr. Karen Marais, Chair

School of Aeronautics and Astronautics

Dr. Ali K. Raz

School of Aeronautics and Astronautics

Dr. Senay Purzer

School of Engineering Education

Approved by:

Dr. Gregory A. Blaisdell

Li Beirut

ACKNOWLEDGMENTS

Thank you to my family for your unconditional love and support. I am infinitely lucky and grateful to have you.

Thank you to my parents. Wajih for always trusting me and my decisions, for sending me the funniest jokes that got me through my roughest days; Laure for being the role model that you are, for inspiring me to be the strongest and best version of myself possible.

Thank you to my sisters. Tina for constantly pushing me and helping me every step of the way; Celine for believing in me even (and especially) when I could not.

Thank you, Wadih for all the Panda nights that helped preserve my sanity, and to Gaby, Line, Mira, Leo and John, for being my home away from home.

Many thanks to Dr. Ali Raz and to Dr. Senay Purzer for your valuable guidance, feedback and encouragement throughout this research.

Lastly, thank you, to my advisor Dr. Karen Marais, for taking a chance on me two years ago. Thank you for your continuous support, incredible patience and commitment to both my academic success and my wellbeing. You have made me a better researcher and a stronger person.

ACKNOWLEDGMENTS

This work is partly supported by the National Science Foundation under Grant No. 1350589.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

This research addresses the need to advance systems engineering education, by assessing current undergraduate systems engineering programs in the US relative to the needs of the industry.

We extracted over 300 expressions relevant to the systems engineer's duties from six sources. We chose sources that address the variety in how people define "systems engineering", the evolving nature of the field, its practical aspect and the lessons learned through experience. We used these expressions to write 35 needed learning outcomes that should be taught to systems engineering students. The outcomes fall under six broad categories relating to requirements management, solution selection and implementation, system architecture and modeling, system performance evaluation, V&V activities and project management. We then looked at what existing undergraduate systems engineering programs are teaching and extracted each program's current learning outcomes. We compared each program's current outcomes to the industry-based needed outcomes to determine whether students are being taught what they need to know.

We learned that the duties of systems engineers are not uniquely defined and prioritized by the six sources, and that academic programs do not all teach the same outcomes. We found that all current undergraduate systems engineering programs in the US are preparing students to meet at least some of the needs of the industry, such as to "Identify stakeholder needs", "Develop high-level system architecture" and "Estimate cost", but that most programs do not teach students how to "Select optimal concept" or how to "Analyze system resilience".

This work motivates the need to investigate potential gaps in systems engineering education and to determine how well we are preparing students to meet the needs of the industry.

1. INTRODUCTION

1.1 Motivation

As the complexity of systems increases, so does what can go wrong with them. The U.S. Aerospace and Defense department, for example, has been handling increasingly complex systems over the years. The percentage of projects that incurred a total cost overrun has increased from 28% to 48% in 12 years (Lineberger & Hussain, 2016). The increasingly large, complex, and inter-disciplinary nature of systems and the current trends in systems failures have both driven demand for systems engineers up. Universities have responded by creating standalone programs in systems engineering, or incorporating systems engineering principles, processes and practices into existing programs. But how effective are these efforts in preparing systems engineers to the problems they encounter in the industry? Are we teaching systems engineering students what they need to learn, and are we teaching it well?

1.2 Research Questions and Approach

In this research, we assess current educational efforts in systems engineering at the undergraduate level, relative to the needs of the industry. More specifically, we answer three questions:

i. What should we be teaching systems engineering students? *(CHAPTER 2)*
ii. What are we currently trying to teach them? *(CHAPTER 3)*
iii. Are we teaching them what they need to be taught? *(CHAPTER 4)*

To answer these questions, (i) we generate a set of learning outcomes that should be taught to students, (ii) we identify the learning outcomes currently being taught to students, and (iii) we compare the two sets of learning outcomes.

1.3 Systems Engineering and Engineering Education

This research falls under the systems engineering education field, so it draws from both the Systems Engineering and the Engineering Education disciplines. Before we discuss relevant previous efforts or dive into our research questions, we define key terms used throughout this research. We explain how we view "systems engineering" and "learning outcomes", two terms defined and used differently across each community.

1.3.1 Systems Engineering relative to Design and to Project Management

Systems Engineering (SE) is a relatively recent but rapidly evolving field, in terms of scope and definition. While efforts have been made to address the "confusion over what is systems engineering, and what is a systems engineer" (Mar, 1997), the literature consistently mentions a lack of consensus on systems and systems engineering definitions. But because we are more interested in the tasks that go into systems engineering processes, we do not look at what systems engineering is, but we look at what it does. A good way of understanding what an emerging field does, is to look at how and why it emerged (What did people do before SE, and why wasn't it working anymore? What problems were SE processes designed to address?), and what distinguishes it from closely related fields, like design and project management.

Here, we refer to two systems engineering textbooks, published in 1992 and 2011 respectively. Although these two textbooks were published almost two decades apart, they describe systems engineering similarly in relation to design: systems engineering emerged from the design field to address issues which emerged with the increasing complexity of systems. When physical systems increased in complexity and were no longer designed by a single individual but rather a design team, a new set of issues emerged. Initially, large design issues were decomposed into smaller subsystem design issues and the system became a collection of designed subsystems. But designing and developing a collection of functional valid subsystems did not necessarily result in a coherent, functional system which fulfills its intended purpose and required capabilities (Page, 1992). When the subsystems have dependencies, an integrative approach is needed to take interactions within the system into account, and between the system and the environment in which it needs to operate. As separate design teams started working on the different subsystems, more incompatibility issues arose, and the need for structured requirements and interface management emerged.

The development of systems engineering practices was accelerated by World War II. The rate of technological advances accelerated as countries raced to design and develop innovative, complex systems. Such projects combined several technical disciplines and required high levels of organization and efficiency to meet tight schedules. These new types of design challenges required a transdisciplinary and integrative approach, a systems engineering approach. However, the goal of systems engineering processes is not to replace the design process, but to guide and complement it.

The output of the systems engineering process includes “information concerning the specifications or architecture for the product or service that will ultimately be manufactured, implemented, installed”. Design is defined as “the creative process through which system products, presumed to be responsive to client needs and requirements, are conceptualized or specified, implemented, and maintained” (Page, 1992).

In other words, systems engineering efforts guide the design process by providing requirements that the product must satisfy; design efforts are responsive to and support the systems engineering life cycle. Systems engineers also contribute to the system design, by leading the concept development and high-level system architecture satisfying the needs of the customer; design engineers are responsible for specific system production and implementation (Kossiakoff et al., 2011).

Systems engineers also work along project managers to ensure the design engineer stays within program schedule and cost. Systems engineers contribute to other aspects of project management, such as setting the engineering effort’s objectives, guiding its execution, evaluating performance and developing corrective actions (Kossiakoff et al., 2011) . They are not, however, concerned with operations planning, cost, schedule, performance, risk monitoring and control (SEBoK Authors, 2020).

1.3.2 Engineering Education

For our purposes, a learning outcome is a statement describing what a student should be able to do at the conclusion of a course. To ensure that a learning outcome is observable and measurable, it is usually written using verbs from Bloom’s Taxonomy.

ABET is an organization that accredits college and university programs. ABET-accredited engineering programs must cover a certain set of general engineering learning outcomes (the General Criteria), in addition to a set of discipline-specific requirements (Program Criteria). For example, an engineering management program must satisfy the General Criteria to be accredited, but its curriculum must also “prepare graduates to understand the engineering relationships between the management tasks of planning, organization, leadership, control, and the human element in production, research, and service organizations" (among other Program Criteria). However, for "Systems and Similarly Named Engineering Programs", ABET does not require program-specific criteria beyond the general engineering criteria (Criteria for Accrediting Engineering Programs, 2019 – 2020, 2019). This is one of the reasons the literature mentions a lack of guidance on what to teach systems engineering students and the need to address systems engineering education.

1.4 Systems Engineering Education

Previous efforts have been made to advance systems engineering education, by identifying what should be taught to systems engineering students. Peng and Ferris (2006) developed a framework for systems engineering education within a "traditional" engineering discipline (vs a stand-alone SE degree program). The authors built the stream of studies around what they consider to be the SE educational needs at the undergraduate level: a "general introduction to the field leading to the appreciation of systems issues in the development of products and systems" and "an introduction to tools and techniques for performing systems engineering tasks". They based their framework on Taiwan's national needs (as the country's economy had experienced significant growth and changes), on the Hitchins model of systems engineering (2000), and on educational theory (e.g. Bloom's Taxonomy).

While this study offers an answer to our first research question (i.e. what we should teach systems engineering students), our approach is different in several ways. First, we are not designing a program, but assessing existing ones using metrics we developed. Second, the scope of the proposed framework is limited to the "character" of each academic year, whereas our research answers the question with a set of more specific, courses-level learning outcomes. Third, we base our answer (the learning outcomes) on a different set of requirements, which are more industry-based.

We choose to answer this first research question with a set of learning outcomes because most educational programs in the US are outcome-based. In order to compare an educational program's offerings to the needs of the industry at a later stage of this research, we first need to express both in a common "language" and similar "format". We use chapter 2 to translate the industry needs into "needed learning outcomes" and chapter 3 to extract the "current learning outcomes” from academic programs. Chapter 4 compares the two sets of outcomes to determine whether there is a gap between the industry’s needs and the academia’s offerings.

Previous research also assessed the effectiveness of a systems engineering program relative to the needs of a specific company, which co-developed the program (Goodlass, 2004). To address the fact that “academia was not producing what industry wanted”, Loughborough University developed a systems engineering program, partly based on an industry partner’s need for graduates able to “perform an integrating role within the company”. They then assessed the effectiveness of the program by looking at several indicators, such as program enrollment numbers, students’ academic performance, and post-graduation plans. Students in the program performed well, with an average of 80% of each cohort awarded first class honors or upper second-class degrees. Enrollment numbers and competition to recruit the graduates both increased over the years. The company also compared graduates from this program to other engineers they have employed. They found that the program’s graduates exhibited no disadvantage compared to their peers, but seemed to possess advantages over them, like the ability to manage complex interactions and dependencies between people, products and processes. They also possessed a breadth of knowledge spanning several disciplines and a system thinking capability. High-tech and consultancy companies are increasingly seeking these graduates, and managers hiring them are satisfied with their performance. Loughborough University’s systems engineering graduates seemed to possess certain SE advantages over graduates from other engineering disciplines. Unlike other employees, the graduates of this program received a formal education based on the company’s specific needs and what they wanted them to learn. For this program and for this particular industry partner, the answers to our third research question seems to be “yes”: the program did teach students what the industry partner needs them to learn, and it did it well (students were competent in required SE skills). The program seems effective in training systems engineers relative to the needs of the specific industry partner.

However, the systems engineering needs of Loughborough University’s industry partner might be unique or not completely representative of the rest of the industry. Systems engineering is a relatively recent field which is still evolving rapidly in terms of scope and definition. As such, a systems engineer’s role can vary across companies and industries. Our research adopts a broader approach to capture the wider needs of the industry, by referring to a variety of sources. In addition, we seek to evaluate the current state of systems engineering education, which requires looking at several programs. Another difference between our approach and the one employed by Goodlass, is the way we measure program effectiveness. Here, we assess the state of systems engineering education by looking at a program in terms of its learning outcomes and comparing these outcomes to the industry-based set. This approach is more quantitative, as the comparison assigns scores to each program.

2. GENERATING NEEDED LEARNING OUTCOMES

In this chapter, we answer the first research question and determine what learning outcomes systems engineering students should be learning. We base our answer and outcomes on the needs of the industry, which we identify by looking at a variety of sources.

Our sources cover several perspectives on systems engineering tasks and address gaps in systems engineering education mentioned in the literature. The first two (INCOSE Systems Definitions and Activities and Systems Engineering Published Research) address the fact that systems engineering has several and evolving definitions; sources Three and Four (NASA Systems Engineering Handbook and Job postings) capture the systems engineering process in practice; and sources Five and Six (Interviews with Systems Engineers and Causes of Failures) offer insight on some of the lessons learned in the industry.

From each source, we extract the expressions related to the duties of a systems engineer. We then group and filter the extracted expressions to write a set of industry-based needed learning outcomes.

We do not rank our learning outcomes by importance. This is because systems engineering tasks are not uniquely defined, and each source might consider a different set of tasks to be more important than another. We also do not order them according to where the task they correspond to fall in the systems engineering process, because an outcome needs to be taught regardless of when the task needs to be achieved.

2.1 Identifying the needs of the industry

2.1.1 INCOSE Systems Definitions and Activities

As previously mentioned, systems engineering is a developing field. Because it does not have a single, constant and universally agreed-on definition, the International Council on Systems Engineering, INCOSE, established The Fellows Initiative on System and Systems Engineering Definitions to review some current systems engineering-related definitions and "recommend any changes necessary to align the definitions to current practice" (Systems and SE Definitions, n.d.). Their "Systems Engineering and System Definitions" document is based on 350 comments and suggestions from the systems engineering community and lists the ten activities at the core of systems engineering. These activities describe what a systems engineer should be able to do according to INCOSE. From each activity, we extract key words and key concepts, as illustrated in the following example:

Activity 1: "establishing stakeholders' success criteria and concerns, and defining actual or anticipated customer needs and required functionality, early in the development cycle, and revising them as new information is gained and lessons are learned".

We first divide the activity into smaller "chunks", each pertaining to a single idea or action:

- Idea 1: establishing stakeholders' success criteria and concerns
- Idea 2: defining actual or anticipated customer needs and required functionality, early in the development cycle
- Idea 3: revising them as new information is gained and lessons are learned.

Since at a later stage we will be looking at each idea individually, each idea needs to be complete on its own and we need to be able to understand it without referring to other ideas. We replace words that refer to external elements by what they represent. For example, "them" in activity 1's third idea refers to "customer needs and required functionality", so we rewrite idea 3 as: "revising customer needs and required functionality as new information is gained and lessons are learned."

As we mentioned in the introduction of this chapter, we are looking at the tasks that a systems engineer does, regardless of when they do them in the systems engineering process. We remove time and context-related indications to obtain our first three extracted expressions:

- establishing stakeholders' success criteria and concerns
- defining actual or anticipated customer needs and required functionality
- revising customer needs and required functionality.

We repeat the process for the remaining nine activities. The words and expressions obtained are shown in Appendix A.

2.1.2 Systems Engineering Published Research

"Surprisingly, in a recent and evolving field, there are already references to "the old SE" (or the traditional, the classical, the ordered) and "the new SE"" (Ramos, Ferreira, & Barcelo, 2012). The discipline of systems engineering has experienced rapid developments in the last few years, but we are interested in the current needs of the industry and therefore the current systems engineering focus. Research papers provide us with a way of monitoring current trends, popular research areas, and key developments.

We look at papers published in INCOSE conferences and journals since they include content focused on systems engineering. We consider the "Best Papers" featured by INCOSE in a virtual issue, selected precisely to "provide some insight into the latest research and developments in systems engineering", as stated on their website. The key and most commonly used words in these papers can help us identify important themes in the systems engineering process, field and profession.

Using an online word counter, we obtain the most used words in the abstract of each paper. In Appendix B, we list words and bigrams that occurred at least three times and trigrams that occurred at least twice in the abstract of a paper.

2.1.3 NASA Systems Engineering Handbook

The first/previous two sources informed us on systems engineering features, as defined from a theoretical/conceptual and research point of view.

Next, we consider what goes into the systems engineer's role in practice, or how systems engineering is practiced. We need to identify how the industry translates and adapts systems engineering theories into a process that is followed in practice.

The National Aeronautics and Space Administration, NASA, is an agency that deals with some of the most complex systems in the world, and uses systems engineering processes to do so. Their *NASA Systems Engineering Handbook* provides details on the systems process that is to be followed in the development and implementation of all projects, and describes systems engineering best practices (Shea, 2020). It was published to "bring the fundamental concepts and techniques of systems engineering to NASA personnel in a way that recognized the nature of NASA systems and the NASA environment". It adopts a top-down approach, starting with an overview of the systems engineering process and the steps of that process, before going into the details of each step based on how NASA does it. Chapter One consists of an introduction on the handbook purpose, scope and depth; Chapter Two goes over the fundamentals of systems engineering; chapter three describes the phases of the NASA program/project life cycle; and chapters four to six describe the details of each step of the Systems Engineering Engine in terms of input, activities and output and with examples relevant to NASA.

The handbook considers systems engineering "as it should be applied throughout NASA" specifically and "provides perspectives relevant to NASA and data particular to NASA". As such, the detailed steps described in the second part of the handbook are specific to and aligned with NASA's policies and might not be representative of how systems engineering is practiced in other organizations. We therefore consider Chapter Two in our analysis, more specifically its introduction and first section, "The Common technical processes and the SE Engine". This first section presents an overview on the processes which make up the SE Engine (Figure 2.1) and describes what each set of processes is used for.

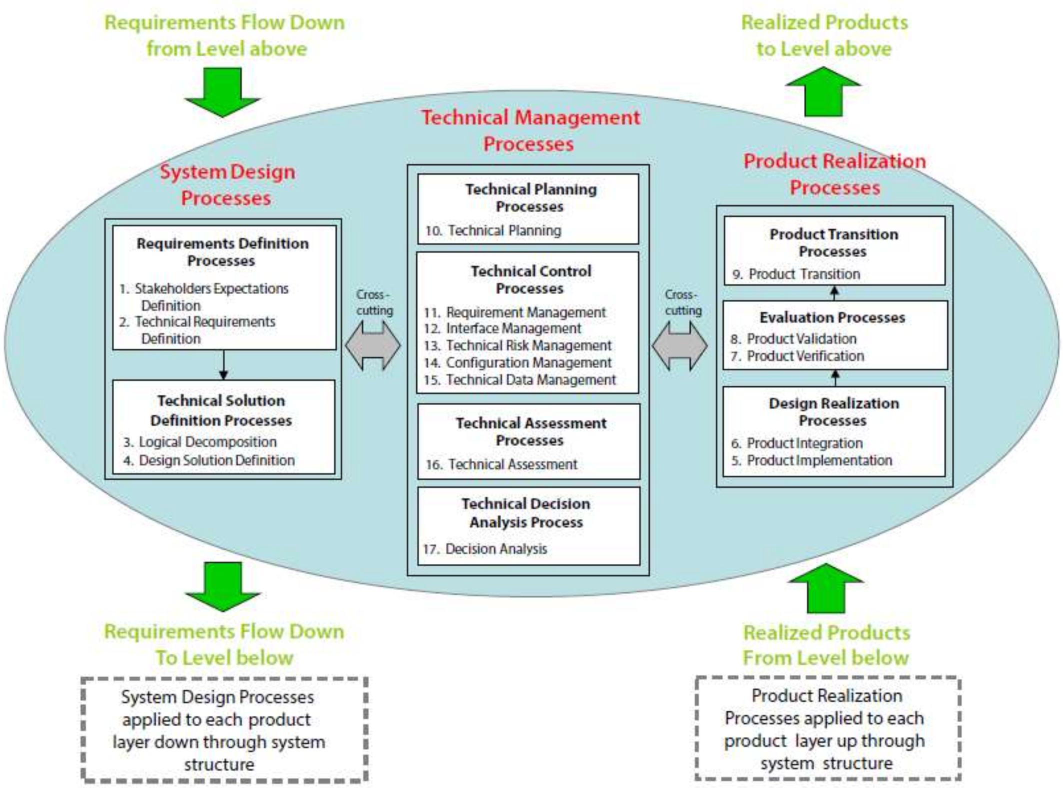


Figure 2.1 The NASA Systems Engineering Engine (NASA, 2020)

From this section, we extract the name of the 17 processes also shown in the figure ("Stakeholder expectations definition", "technical requirements definition", etc.) and their purposes (from the descriptive text).

The introduction of chapter two of the NASA SE Handbook (three pages written between the titles of chapter 2 and section 2.1) describes the responsibilities of a systems engineer that are not part of the SE Engine, such as the "the prime responsibility in documenting many of the technical plans, requirements and specification documents, verification and validation documents, certification packages, and other technical documentation". We use a process similar to the process followed in Section 2.1.1 to extract expressions related to the role the systems engineer plays in technical planning, project management and system architecture development.

The expressions extracted from Chapter two of the NASA SE Handbook (both the introduction and the SE Engine) are listed in Appendix C.

2.1.4 Job postings

A way of determining the industry's needs in terms of systems engineering is to look at job postings. Systems engineering job postings list the qualifications that make a systems engineer a good candidate and the skills they need to perform the job. They also list the systems engineer's responsibilities and the tasks they are expected to know how to complete.

To determine which skills companies expect systems engineers to learn at least in part at school as opposed to acquiring through experience, we narrow down our search to entry-level positions, open to applicants who have just finished their undergraduate degree and have no industry experience. These candidates' systems engineering knowledge comes mainly from their undergraduate studies.

Some positions labeled as Systems Engineering positions are software focused (Computer systems). We do not include them in our analysis.

We look for job postings with the following criteria:

1. Job title has "Systems Engineer".
2. Qualifications do not require graduate studies (BS is enough).
3. Position is entry-level (does not require industry experience).
4. Description of position matches our definition of systems engineering and not the software-focused systems engineering (computer systems).
5. Position is based in the USA.

We use two search engines, Google and LinkedIn Jobs, to search for "systems engineering entry level".

We went through the descriptions of the postings, select the first five positions that meet our criteria from each search engine and make sure the two sets are mutually exclusive (no repeated postings). We notice that most postings which meet our criteria are for positions at aerospace or defense companies.

To obtain a general perspective on systems engineering duties, we select postings for companies from four different industries: aerospace and defense, medical technology, management and information technology consulting, and technology.

From each selected posting, we extract the list of “key accountabilities” or “responsibilities”. The expressions extracted from our nine selected job postings can be found in Appendix D.

Note: While most jobs found during our search have “systems engineer” as title, some have more specific titles and specialized responsibilities. These types of positions are mostly found at aerospace companies. Boeing, for instance, was hiring “Systems Engineers” of different levels (entry level, associate, mid-level, seniors, experienced), but also a “Systems engineer – MBSE", a “System validation and verification engineer”, a “Systems engineer: technical proposal team”, etc.

2.1.5 Interviews with Systems Engineers

> “The improvement that should be expected by condensing and amplifying the lessons of experience are not defined.” (Armstrong & Wade, 2015)

Next, we consider previous research which identified lessons learned within the industry and shortcomings of current systems engineering practices and education. Aloisio (2016) conducted interviews with experienced and practicing systems engineers from a large-scale aerospace company, as part of efforts to gain insight into problems that contribute to project failures and to inform systems engineering education.

Their four interview questions that collect “criticisms on education” are:

1. What is your general academic background and what schooling or other training did you receive before becoming a Systems Engineer?
2. Have you noticed any areas systems engineers struggle with the most when they are first hired?
3. Do you look for certain traits when hiring systems engineers?

4. Have you noticed any changes in systems engineering practices over time at your company?

We consider the engineers' answers to questions 2 and 3, because they "highlight deficiencies in systems engineering education" and "provide contrast to what experienced systems engineers value versus what systems engineering educators value".

We use the process described in Section 2.1.1 to extract expressions related to what systems engineers need to be able to do when they start out in the industry. The extracted expressions from Interviews can be found in Appendix E.

2.2 Translating industry needs into needed learning outcomes

The process we described in Section.2.1 allowed us to extract over 300 expressions related to the duties of a systems engineer. We copy these expressions to a single Excel sheet, and sort them by their source. We filter these expressions twice. First, we sort them into broad categories.

Figure 2 illustrates how we organized our data to obtain our first broad category. We start by taking the first extracted expression, "Stakeholder", which was mentioned several times in a paper from 2018. We then go through all the other extracted expressions, from all sources, and look for expressions that relate to "Stakeholders", such as "Stakeholder requirements definition" and "Customer needs".

During this process, we notice that stakeholders are mentioned in the systems engineering process because their needs set some of the required capabilities of the system, or the requirements. More specifically, systems engineers interact with stakeholders to identify the requirements (requirements elicitation). We select stakeholder and requirement-related expressions (**bolded** in Figure 2.2) and group them together, since they are closely related and are part of the same systems engineering step. At this stage, we do not consider what should be done with requirements i.e. "identifying requirements" or "flow-down requirements" both belong to the "Requirements" category.

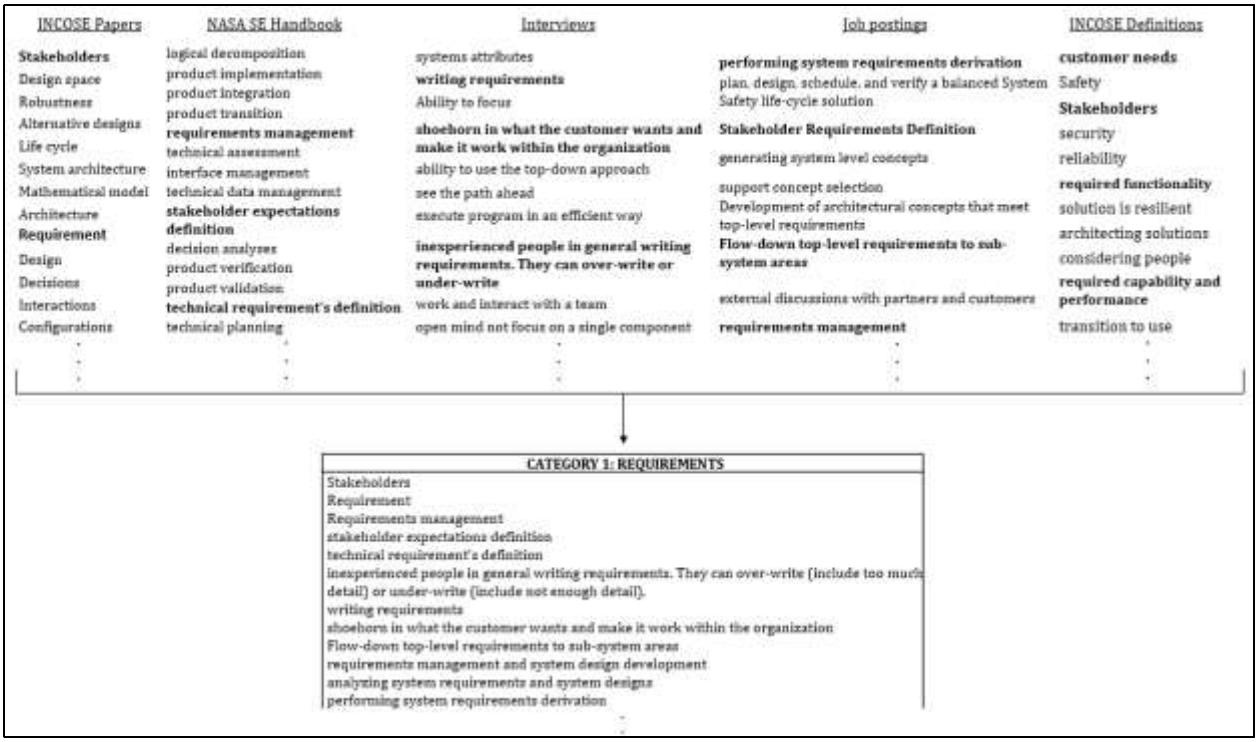


Figure 2.2 Generating needed outcomes: example of first filtering

Next, we go through the list of extracted expressions under the Requirements category, to understand what kind of actions and tasks are involved. We find that systems requirements need to be generated based on stakeholder identified needs, then flowed-down or propagated to lower-level subsystems and components. Systems engineers need to also be able to understand and analyze requirements. Our first category therefore relates to the identification, generation, propagation, and analysis of requirements.

This first "rough" filtering produced the following six categories:

1. Requirements identification, generation, propagation, and analysis
2. Solution architecture and modeling (interactions, interface, etc.)
3. Design space definition and concept selection
4. Risk management and system performance parameters analysis
5. Verification, validation and testing activities
6. Project management and planning, technical writing and teamwork.

In the second filtering step, to further refine the classification, we repeat the process, this time considering one category at a time, as shown in Figure 2.3. Fom each category, we obtain five or six groups of closely related expressions. Each group of expressions is relevant to a systems engineering task and is used to write an industry-based learning outcome which contains all of its ideas. To ensure our learning outcomes are measurable and observable, we write them using verbs from Bloom's taxonomy.

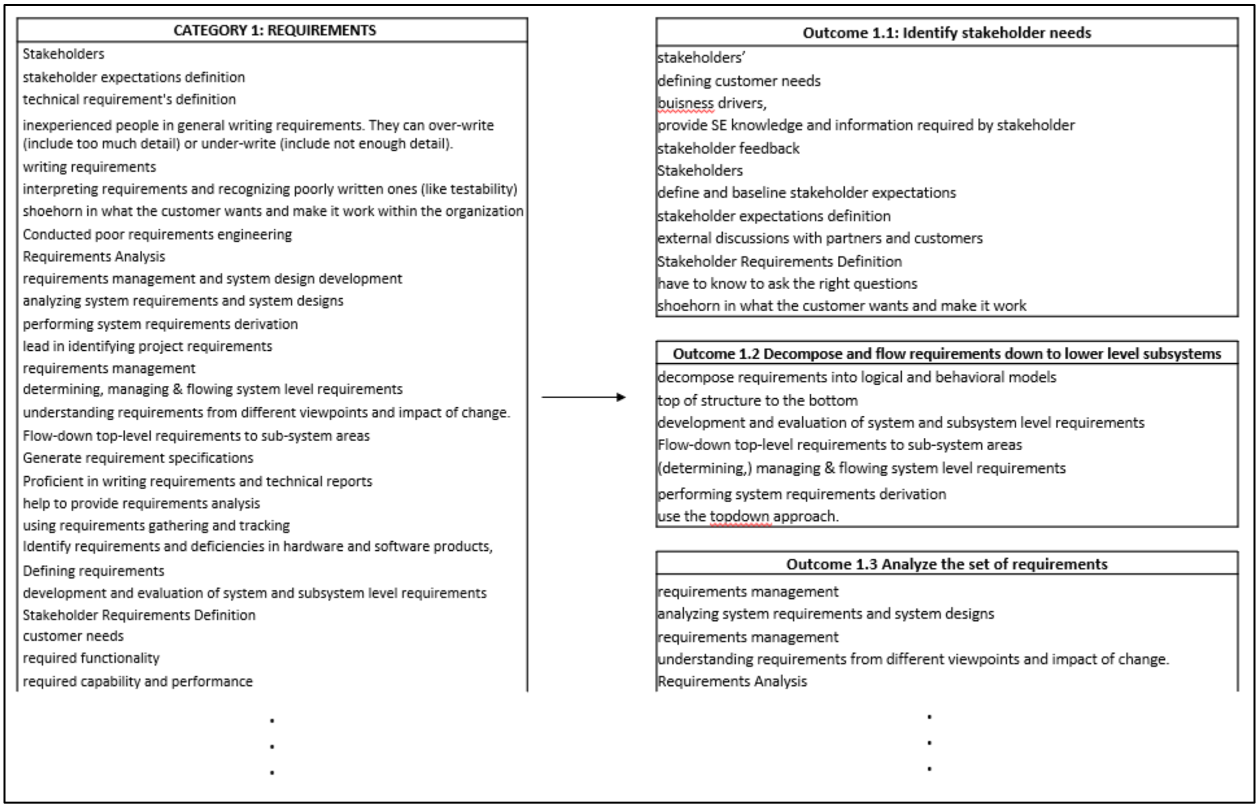


Figure 2.3 Generating needed outcomes: example of second filtering

<u>Note</u>: Eleven extracted expressions from Research Papers, such as "systems", "field", "engineering" and "different", remained unclassified because they are too broad or irrelevant.

2.3 The set of needed learning outcomes

Category 1. Requirements identification, generation, propagation and analysis

Outcome 1.1. Identify stakeholder needs

Outcome 1.2. Analyze requirements

Outcome 1.3. Write and document requirements for traceability

Outcome 1.4. Identify requirements at system level

Outcome 1.5. Generate system concept of operations

Outcome 1.6. Decompose and flow requirements down to lower level subsystems and components

Category 2. System architecture and modeling

Outcome 2.1. Decompose system to create subsystems

Outcome 2.2. Integrate subsystems into a coherent system

Outcome 2.3. Develop system architecture

Outcome 2.4. Define and manage interfaces within system and between system and environment

Outcome 2.5. Model system

Outcome 2.6. Identify interactions and dependencies between subsystems

Category 3. Solution selection and implementation

Outcome 3.1. Determine solution space

Outcome 3.2. Compare alternatives and design trade offs

Outcome 3.3. Select optimal concept

Outcome 3.4. Manage different configurations

Outcome 3.5. Apply decision-making and analysis tools

Outcome 3.6. Manage system across its lifecycle including implementation, transition to use, operation, maintenance, end-of-life processes

Category 4. System performance parameters analysis

Outcome 4.1. Establish effectiveness and performance metrics

Outcome 4.2. Assess system effectiveness and performance

Outcome 4.3. Manage risk

Outcome 4.4. Manage system safety and reliability

Outcome 4.5. Manage system resilience

Category 5. Verification, validation and testing activities

Outcome 5.1. Perform system performance validation

Outcome 5.2. Conduct verification strategies on system and components to ensure requirements and operational goals are met

Outcome 5.3. Define and develop test plans

Outcome 5.4. Execute tests plans and procedures

Outcome 5.5. Evaluate V&V test results

Outcome 5.6. Document test plans, protocols and results

Category 6. Project management and planning, technical writing and teamwork

Outcome 6.1. Manage project technical planning

Outcome 6.2. Produce technical documentation of the project

Outcome 6.3. Communicate in organized, clear and concise manner

Outcome 6.4. Manage a multi-disciplinary team

Outcome 6.5. Estimate cost

2.4 Mini Data Exploration

In the previous three sections of Chapter 2, we identified the needs of the industry in terms of systems engineering and translated them into a set of learning outcomes, to answer "What should systems engineering students be taught?". In the process, we generated a set of data, consisting of 313 expressions related to the systems engineer's duties, extracted from a variety of sources. Following the steps described in section 2.2, we divided this set of data into 6 categories and then 39 groups of expressions. We used these groups to write 39 learning outcome, representing the needs of the industry, as viewed by our six sources.

But do our sources *view the needs of the industry similarly?*

We took the decision to include several and various sources early on, because the literature consistently mentions a lack of consensus on the definition and scope of systems engineering, partly due to the relative recent emergence of the field, its rapid evolution, and its cross-cutting interdisciplinary nature.

As part of their Initiative on System and Systems Engineering Definition, the INCOSE Fellows deployed a survey investigating members perception of systems engineering. They looked at the agreement in responses to the 15 survey questions, which can be aggregated into five broader questions:

1. What is Systems Engineering (SE)?
2. What are the defining characteristics of SE?
3. What is the scope of SE?
4. Who should have knowledge of SE? and
5. What does the future of SE look like?

Three out of the four survey questions which together answer “What is Systems Engineering?” did not have a single response with agreement above 75% (Arnold, Jackson, & Sillitto, 2017). However, only one of the six survey questions related to the scope of systems engineering did not have responses with an agreement above 80%; all remaining survey questions did. The survey results suggest that there are many systems engineering definitions, but that most of them describe “different aspects of the same thing, and at different levels of abstraction, rather than different things” (Jackson, 2018). A similar survey investigating different “system” definitions found a larger variety in responses and identified seven different worldviews on “system”.

The idea of a lack of consensus on what a system is, what systems engineering is, or what a systems engineer does, is mentioned in the premise of several papers, but is rarely (dis)proved as thoroughly and quantitively as done by INCOSE’s initiative, or even referenced; in most cases, the idea seems assumed, presented as a fact.

Here, we look at whether the needs of the industry are uniquely perceived across our six sources, or whether there is a disagreement on what constitute the duties of a systems engineer across the community. We can do so by taking a closer look on the 35 sets of expressions, which each refer to a systems engineering task and needed learning outcome. We examine the composition of each group. If each source had at least one expression in each group, this would mean that all six sources referred to the same set of systems engineering duties.

We use a simple color coding to assign each cell containing an expression a color coding, depending on the expression’s source. Figure 2.4 shows the first three groups of expressions, color-coded by source.

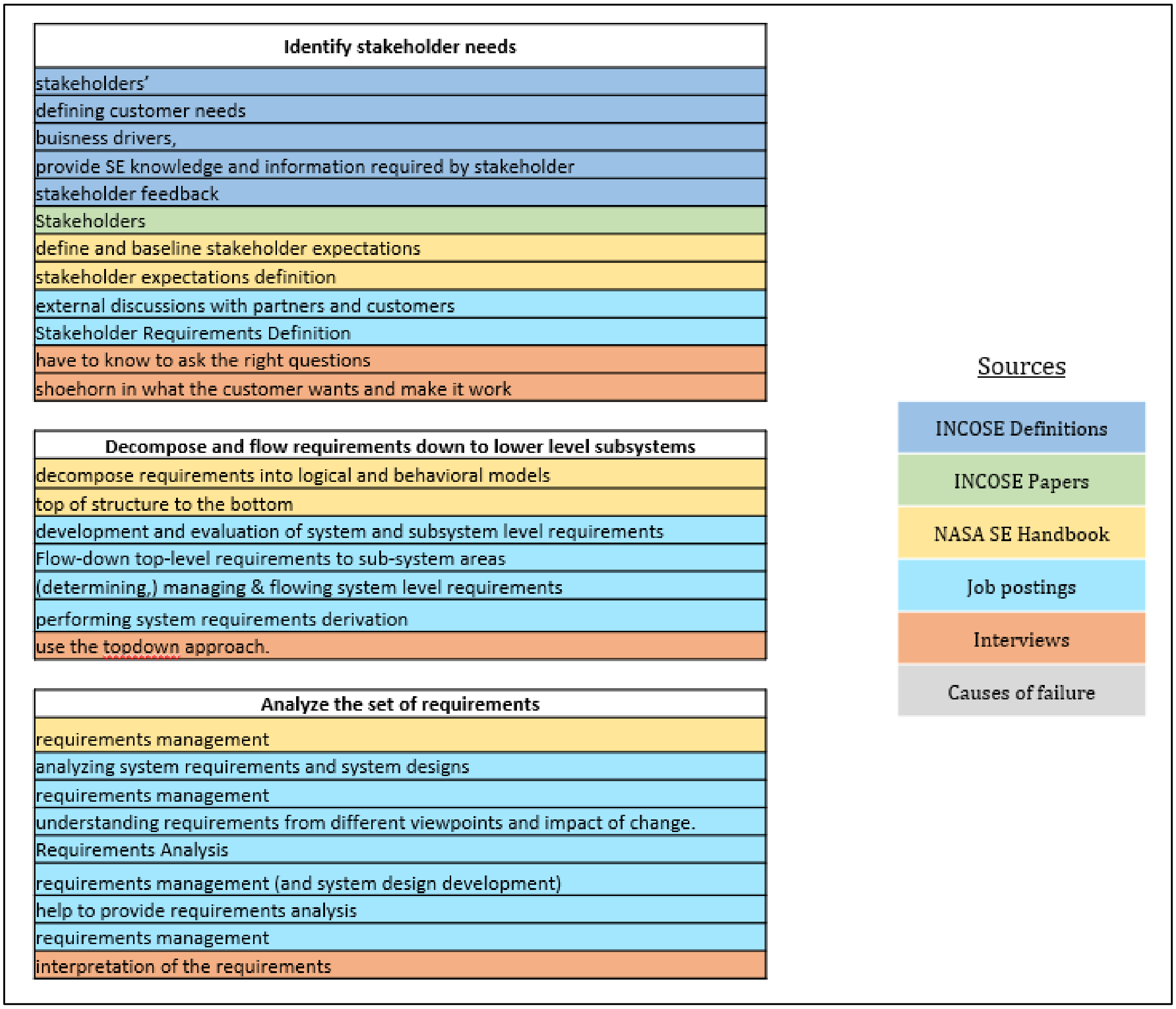


Figure 2.4 Sample of needed outcomes, color coded by source

While the first group of expressions suggests that every source does refer to every learning outcome, the next two groups do not. The color coding makes it clear from the first few formatted groups that every source does *not* refer to every learning outcome. While some difference was somewhat expected, the color coding reveals a complete divergence in compositions: the first group has expressions from five different sources (all but the causes of failure), while groups Two and Three had expressions from the same three sources, with the same weights. By visually examining three outcomes, we can already claim that: Some of the needs of the industry are recognized by most of the six sources, others by only one or two.

This small observation prompts a series of other questions. Do more outcomes have compositions close to outcome 1's or to outcome 2's? Which outcomes are commonly agreed on? Was a certain type of outcomes consistently mentioned by most sources? Did each source focus on a specific category of outcomes? Are there different worldviews on the systems engineer's duties?

Here we pause for a moment to make a disclaimer: This research does not aim at providing an answer to "Are there different worldviews on the needs of the industry in terms of systems engineering?". However, it takes into account the possibility of the answer being "yes", by referring to six sources of different types to capture the needs of the industry as broadly as possible. Our research method (and therefore choice of sources) was not designed to check for specific worldviews on the systems engineer's duties and so the analysis, interpretation and discussion presented in this section might not give a complete picture of the different views on a systems engineer's duties. It might, however, reveal potential patterns or shed light on areas worth investigating/needing further investigation.

To see whether more observations can be made, we analyze the composition of the remaining 35 groups. To be able to draw stronger, more objective conclusions, we quantify the compositions of the groups.

In this section,

- We say "a source contributes to an outcome" or "a source covered an outcome" if that source has at least one expression in the group corresponding to the outcome.
- The "number of contributions of a source to an outcome" is the number of expressions from that source in the group corresponding to the outcome.

Applying this terminology to the outcomes illustrated in Figure 2.4, we can write the following statements:

- Almost all sources covered the outcome "Translate stakeholder needs into requirements".
- INCOSE Activities contributed five times to Outcome 1.
- Job Postings was the main *contributor* to Outcome 3.
- Sources contributed to the first three outcomes very differently.

We count how many times each source contributed to each learning outcome and store the results in a table.

For example, for outcomes 1 to 3 (shown in Figure 2.4), we obtain the rows shown in Table 2.1.

Table 2.1 Contribution of each source to the first three needed outcomes

	INCOSE	PAPERS	NASA	JOBS	INTERVI EWS	CAUSES
Identify stakeholder needs	5	1	2	2	2	0
Decompose and …	0	0	2	4	1	0
Analyze requirements	0	0	1	7	1	0

We use color coding once again to visualize how much each source contributed to each outcome. We choose a green-yellow-red color scale, where green represents the cell with the highest number of contributions and red the cell with the lowest. But what is considered a "high" contribution to an outcome? The answer depends on the source.

Two contributions from the "NASA SE Handbook" to an outcome is pretty average, considering we have a total of around 56 expressions extracted from that source, spread across 40 outcomes; but two contributions from "Interviews" to an outcome might be more meaningful, since we only extracted 30 expressions from it. If we were to apply the same color scale across both columns, these two cells would have the same color and therefore represent "equal" contributions. In the case of the sample presented in Table 2.1, they would be represented by the same shade of orange indicating "low" contribution (values in table range from zero to seven).

We therefore use a different scale for each column, relative to the source's total number of extracted expressions, as shown in Figure 2.5. Table 2.2 below shows how the sample presented in Table 2.1 would look, with this scale applied on each column.

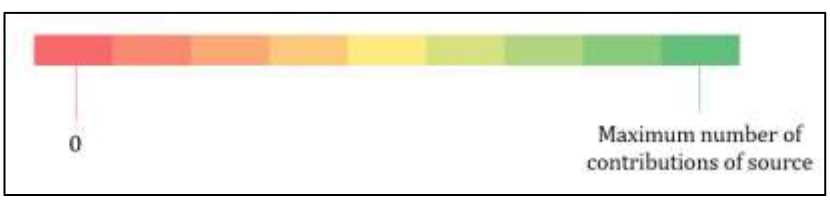


Figure 2.5 Scale used for data exploration

Table 2.2 Contribution of each source to the first three needed outcomes, color-coded

	INCOSE	PAPERS	NASA	JOBS	INTERVIE WS	CAUSES
Identify stakeholder needs	5	1	2	2	2	0
Decompose and …	0	0	2	4	1	0
Analyze requirements	0	0	1	7	1	0

For example, a 2 under Jobs is represented in orange (indicating low-medium contribution), while a 2 under Interviews is represented by green (indicating high contribution). A cell's fill color does not indicate the number of times a source contributed to an outcome. It indicates the weight of a source's contribution, or how often it contributed to that outcome relative to the other outcomes. It shows how important that outcome was to the source.

Color scaling the first three outcome's category/outcome contributions can highlight areas where sources had clearly similar or clearly different levels of contributions. However, it might be difficult to visually compare over 200 numerical entries spread across 35 rows and 6 columns, even if we color-scale them, so we start by looking at how each source contributed to each *category* of outcomes, instead of each individual outcome (Section 2.4.1). In Section 2.4.1, we generate and analyze two tables to check whether a source emphasized certain categories of outcomes, or certain outcomes within a category.

2.4.1 Results, by category

In this section, we see whether a source:

a) covers all outcomes from a certain category
b) does not cover any outcome from a certain category
c) emphasizes a subset of outcomes from a certain category

Checking for the first two cases is relatively straightforward: for each source, we count how many outcomes it covered from each category (Table 2.4). This can tell us whether a source mentioned/emphasized a certain category of outcomes, but not whether it emphasized certain outcomes from a category. To better explain what we mean by that, we take the example of a source which covered 3 outcomes in each of categories 1 and 2. Table 2.3 shows the individual contributions of that source to these outcomes.

Table 2.3 Contributions of a source which covered 3 outcomes in Categories 1 and 2

Outcome	Number of contributions
1.1	0
1.2	3
1.3	1
1.4	0
1.5	1
1.6	1

Outcome	Number of contributions
2.1	1
2.2	0
2.3	1
2.4	0
2.5	0
2.6	1

This source contributed six times to Category 1, and three times to Category 2. This source covered the same number of outcomes in each category, but only really emphasized one outcome from Category 1. Looking at only the number of outcomes a source covered in a category does not tell us if that source emphasized a certain outcome; and looking at only the total number of contributions in a category does not tell us if the contributions were "concentrated" in a single outcome or spread across six outcomes. We need to look at both numbers. For each category, we count the number of outcomes a source covered (Table 2.4) and the total number of contributions (Table 2.5).

Table 2.4 Number of outcomes to which each source contributed, per category

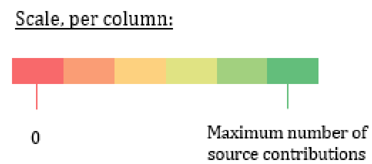
	INCOSE Activities	Research Papers	NASA Handbook	Job Postings	Interviews	Causes of failures
C1	3	2	5	5	4	2
C2	5	4	4	5	2	0
C3	6	5	4	6	0	0
C4	5	3	2	4	0	1
C5	1	2	2	6	0	2
C6	2	1	6	5	5	2

Scale, per row:

0 Number of outcomes per category

Table 2.5 Number of contributions of each source to each category of outcomes

	INCOSE Activities	Research Papers	NASA Handbook	Job Postings	Interviews	Causes of failures	Total
C1	11	2	9	19	9	2	52
C2	8	11	9	22	8	0	58
C3	15	11	12	17	0	0	55
C4	13	11	7	12	0	2	45
C5	3	3	7	28	0	3	44
C6	2	1	12	30	11	3	59



For each source, we look at the numbers of outcomes it covered and of its contributions (we analyze the table, by column):

1. INCOSE Activities:
    a. covers almost all outcomes from Categories 2, 3 and 4 (16 out of 17)
    b. does not focus on outcomes from Categories 5 and 6
    c. seems to also emphasize a subset of Category 1 and a single outcome of Category 5
2. Research Papers:
    a. Covers almost all outcomes from Category 3
    b. Does not focus on outcomes from categories 1, 5 and 6 (five out of 18 outcomes, with low numbers of contributions)
    c. Seems to emphasize certain outcomes from Categories 2 and 4 (seven out of 11 outcomes, with high contribution numbers)
3. The NASA SE Handbook contributes to all categories, in a relatively consistent manner. It:
    a. Covers almost all outcomes from Categories 6 (with high contribution) and 1
    b. Does not focus on outcomes from Categories 4 and 5
    c. Seems to emphasize some outcomes from Category 3
4. Job Postings has the least variation in the number of outcomes it covers across categories. It:

    a. covers most or all outcomes across all categories.
    b. contributes to Category 4 the least
    c. emphasizes categories 5 and 6
5. Interviews covers the different categories in strikingly different ways. It only covered categories 6 and 1 and a subset of category 2. It contributed to 0 outcomes from categories 3, 4 and 5.
6. Causes of failures did not have expressions related to outcomes from Categories 2 and 3.

We can write the following statements:

1. INCOSE Activities covered most outcomes related to Systems Architecture & Modeling, to Solution Selection and Implementation, and to Systems Performance Analysis; it also covered some Requirements-related outcomes but did not consider V&V Activities and Managerial tasks to be core systems engineering activities.
2. Research Papers emphasized outcomes teaching Solution Selection and Implementation, and some outcomes teaching systems architecture & modeling and systems performance analysis. It did not mention Requirements, V&V and Managerial tasks as much.
3. NASA SE Handbook emphasized requirements and management tasks, in addition to some tasks related to the solution selection.
4. Job Postings covered strongly most outcomes across all categories, with a slight emphasis on V&V Activities and Managerial tasks.
5. Interviews only mentioned tasks related to Requirements and Project Management, but also highlighted the importance of identifying interactions and dependencies within the system.
6. Causes of failures did not refer to tasks concerned with system architecture and modeling, nor to solution selection and implementation.

Now, we look at the numbers per category (we analyze the tables by rows):

- Category 3: is a category not at all mentioned in Interviews and Causes of Failures, but highly mentioned in all others.
- Category 5: is the most mentioned category in Job Postings and Causes of Failures, but the least mentioned in all others.
- Category 6: is the least mentioned category in INCOSE Activities and in Research Papers, but most mentioned in all others

2.5 Summary

In this chapter, we answered our first question "what should we be teaching systems engineering students?" with a set of 35 learning outcomes. To do so, we started by identifying the needs of the industry in terms of systems engineering, as described by six different sources. From these sources, we extracted over 300 expressions relevant to systems engineering tasks (Section 2.1), which we filtered into six categories and then into 35 groups of closely related expressions (Section 2.2). We also examined the composition of these groups of expressions (Section 2.4), and found that not all sources emphasized or even mentioned the same set of systems engineering tasks.

3. IDENTIFYING THE CURRENT LEARNING OUTCOMES

In this chapter, we answer the second research question ("What are we currently trying to teach systems engineering students at the undergraduate level?") by identifying the learning outcomes currently being taught to the students across different programs. To do so, we first choose (Section 3.1) and identify (Section 3.2) a set of systems engineering programs. We then look up each program and the courses that students must take to fulfill its requirements. We extract the current learning outcomes, per program, from the syllabus or description of each course (Section 3.3).

3.1 Program selection criteria

The "needed" learning outcomes we generated in Chapter Two were based on six sources. Of these sources, only one (Job Postings) was specific to undergraduate programs. So our "needed" outcomes include, to a large extent, what any systems engineering program should teach. Our needed outcomes can be used to assess both undergraduate and graduate systems engineering programs.

However, the stream of studies within a graduate program is much more flexible and customizable than within an undergraduate program. Undergraduate programs do allow students some flexibility in choosing electives or concentrations, but most credit hours consist of a fixed set of courses (and therefore outcomes) that all students enrolled in that program are required to learn. Assessing an undergraduate program can therefore be done by assessing the learning outcomes covered by the required courses, or the program's core "current learning outcomes".

Graduate students, on the other end, have much more freedom in choosing courses. Programs do not always have "required courses" that all students take. Here at Purdue, for instance, MS students in the School of Aeronautics and Astronautics majoring in Systems Engineering are required to take three systems courses: any two or three of seven methods courses, with the option to take one of nine context courses. Two students enrolled in the same graduate program, with the same major, can be taking two entirely different sets of courses and might learn mutually exclusive sets of outcomes. Since a graduate program might not have a core and common/required set of learning outcomes, we cannot determine what it teaches the way we do for an undergraduate program. The extraction of current outcomes, and therefore the assessment method and criteria, would be entirely different. In Section 5.2.4, we discuss some requirements that the method should satisfy.

Although our "needed" outcomes can be taught at any level, in this research we only assess undergraduate programs. Next, we identify universities that offer "Systems Engineering" as an undergraduate major or minor.

3.2 Identifying programs

INCOSE and the Systems Engineering Research Center (SERC) at the Stevens Institute of Technology compiled a "Worldwide directory of all systems engineering and industrial engineering academic programs". The directory lists Bachelor, Masters and PhD programs in "Systems engineering", "Industrial engineering" and programs that combine one of the two with another field (e.g., "Industrial and Systems Engineering", "Operations Research and Systems Engineering", "Industrial and Manufacturing Engineering", "Systems Engineering and Engineering Management").

We search the directory with the following filters:

- Disciplines: Systems Engineering
- Level: Undergraduate
- Country: United States

Out of the 369 universities listed on the directory, seven are US universities offering undergraduate "systems engineering" programs: George Mason University, George Washington University, Drexel University, University of Arizona, University of Arkansas Little Rock, University of Virginia, and the United States Air Force Academy.

3.3 Extracting the current learning outcomes

On the website of each selected program, we search the degree requirements for the list of courses enrolled students are required to take. For example, students pursuing a BS in Systems Engineering at the University of Virginia are required to take the following courses and electives:

- APMA 1110 Single Variable Calculus II
- CHEM 1610 Introductory Chemistry I for Engineers
- CHEM 1611 Introductory Chemistry I for Engineers Laboratory
- ENGR 1624 Introduction to Engineering
- STS 1500 Science, Technology, and Contemporary Issues
- Math and Science elective I
- STS 1500 Science, Technology, and Contemporary Issues
- APMA 2120 Multivariable Calculus
- CS 1110 Introduction to Programming
- PHYS 1425 General Physics I: Mechanics, Thermodynamics
- PHYS 1429 General Physics I Workshop
- HSS elective
- APMA 2130 Ordinary Differential Equations
- CS 2110 Software Development Methods
- PHYS 2415 General Physics II: Electricity & Magnetism, Optics
- PHYS 2419 General Physics II Workshop
- SYS 2001 Systems Engineering Concepts
- STS 2xxx/3xxx elective
- Science elective II
- APMA 3080 Linear Algebra
- APMA 3100 Probability
- SYS 2202 Data and Information Engineering
- Technical elective
- HSS elective
- APMA 3120 Statistics
- SYS 3021 Deterministic Decision Models
- SYS 3023 Human Machine Interface
- SYS 3055 Systems Engineering Design Colloquium I
- Application elective
- Unrestricted elective
- SYS 3034 System Evaluation
- SYS 3060 Stochastic Decision Models
- SYS 3062 Discrete Event Simulation
- Application elective
- Unrestricted elective
- SYS 4021 Linear Statistical Models
- SYS 4053 Systems Design I
- SYS 4055 Systems Engineering Design Colloquium II
- STS 4500 STS and Engineering Practice
- Technical elective
- Application elective
- Unrestricted elective
- SYS 4054 Systems Design II
- STS 4600 The Engineer, Ethics, and Professional Responsibility

We then search for the syllabi or description of each required systems course. For the schools with publicly available syllabi, the level of details and the way information is presented in the syllabi differ from a school to the other, so the extraction of their current outcomes is done in different ways, as outlined below. Some schools do not allow people with no affiliation to the university to access their syllabi. These schools often show 100-word course descriptions, briefly describing the purpose of the course and listing the main concepts or methods taught. For these schools, the extraction of current outcomes is also done differently:

- Syllabi with dedicated section for learning outcomes: we extract the current outcomes directly.
- Syllabi with course schedule: we extract the list of topics covered throughout the course
- Course description: we split the text into chunks each pertaining to only one concept, method or action taught. In Chapter 4, we will compare each of these chunks to each needed outcome.

Below, we show examples illustrating how we extracted these current outcomes from each type of course information available.

3.3.1 From course schedule

Figure 3.1 shows the first part of the schedule shown in the syllabus of SYST 101 (Introduction to Systems Engineering), a required course for students majoring in Systems Engineering at George Mason University.

Date		Day	Lesson	Activity
8/29	T	1	Introduction Lec 1- What is an engineer?	Introduction
8/31	TH	2	Common Engineering Terms	Mech Universe
9/5	T	3	Common Engineering Concepts	Equations for Work
9/7	TH	4	Work in different systems	Quiz on Eng Concepts
9/12	T	5	Lec 2 – What is Systems Engineering?	.
9/14	TH	6	Lec 3 Define Needs & Requirements	Download CORE
9/19	T	7	Lec 4 Using CORE	Create Context, Ext system
9/21	TH	8	Lec 5 System Modeling	
9/26	T	9	Lec 6 CORE functional modeling	EFFBD's & Simulation

Figure 3.1 Snippet of SYST 101 Schedule at George Mason University

We look at the column describing what each lesson covers. Lessons One to Four offer a general introduction to the field and to the course and are not used to teach specific learning outcomes. We extract the list of topics covered in lessons Five to Nine in Table 3.1.

Table 3.1 List of current outcomes, as extracted from SYST 101 course schedule at George Mason University

Define Needs & Requirements
Using CORE
System Modeling
CORE functional modeling

3.3.2 From course description

Drexel University requires students minoring in systems engineering to take EGMT 465, Introduction to Systems Engineering. While we could not access the syllabus of the course, we can use its description to extract the outcomes it covers:

> "Determining technical requirements for engineering systems and planning technical product design and requirements. Analyzing the functionality,

> interoperability, and sustainability of new engineering systems. Integrating disparate engineering components for overall system optimization. Planning for testing and evaluation of engineering systems to evaluate conformance with technical requirements. Planning optimized organizational structure for execution of complex engineering programs."

Table 3.2 shows the current learning outcomes of Drexel University's EGMT 465, as extracted from the available course description:

Table 3.2 List of current outcomes, as extracted from EGMT 465 course description at Drexel University

Determine technical requirements
Plan product design and requirements
Analyze functionality of new systems
Analyze interoperability of new systems
Analyze sustainability of new systems
Integrate engineering components for overall system optimization

3.4 Remarks

a) In this research, our goal is not to determine everything that systems engineering programs teach. We want to see whether programs teach the set of "needed" learning outcomes. We therefore extract the current outcomes and do not store, group or process them in a way. In the next chapter, we compare each program's current outcomes to the needed outcomes to check which needed outcomes are included in current programs. We then check whether there are certain outcomes not included in our set of needed outcomes that most programs did teach.

b) To maintain the focus on systems engineering education overall, and not on evaluating specific programs, we will not be referring to the name of the university offering a systems engineering program when assessing it. Instead, we assigned each program a random ID number (from 1 to 7).

c) Our second question, which consists of understanding what we are currently teaching students, can be broken down to two things: what we say we teach and what we are teaching. The syllabi are not a perfect reflection of what is actually being taught. Most syllabi do not contain the details of every lesson, and the course instructor may cover more, less, or different material than indicated in the syllabus. We suggest/discuss ways in which future work can address this limitation in Section 5.2.1, but the scope of this research only considers what instructors say they teach. Our assessment is based on the information available in syllabi and course descriptions.

3.5 Summary

In this chapter, we identified the seven undergraduate systems engineering programs, as listed in the "Worldwide directory of all systems engineering and industrial engineering academic programs". We looked for information on what instructors say they teach in the required courses. We extracted the learning outcomes currently being taught in each program, to answer our second research question "what are we currently trying to teach systems engineering students?".

4. COMPARISON OF NEEDED AND CURRENT LEARNING OUTCOMES

In the previous two chapters, we identified the needs of the industry and what needs to be taught to systems engineering students (needed learning outcomes) and we extracted what is being taught in some systems engineering programs (current learning outcomes). In this chapter, we compare the two sets of learning outcomes for every program considered, to answer our third research question: are we teaching systems engineering students what they need to learn?

4.1 Comparison method

The outcome of the comparison should indicate whether current programs are teaching needed learning outcomes. Each program has its own set of current outcomes, but they will all be assessed relative to the same set of needed outcomes. For every program, we build a table similar to Table 4.1 below.

Table 4.1 Sample comparison set up

		Needed Learning Outcomes				
		Identify stakeholder needs	Decompose requirements	…	Compare alternatives	…
Program current	requirements derivation					
	analysis of alternatives					
	requirements elicitation					
	Flow down requirements					
	…					

Each column corresponds to a “needed learning outcome” and each row corresponds to a program’s “current learning outcome”. We go through the table one row at a time and compare each current outcome to all needed outcomes. If we find that the current outcome matches a needed outcome, then the pair gets a “point”, or a “1”. We then look at how many times a program mentioned each needed outcomes, by summing the points in each columns. For the sample shown in Table 4.1, we obtain the following:

Table 4.2 Sample comparison table

		Needed Learning Outcomes				
		Identify stakeholder needs	Decompose Requirements	…	Compare alternatives	…
Current Program	*requirements derivation*		1			
	analysis of alternatives				1	
	requirements elicitation	1				
	Flow down requirements		1			
	…					
	Program total scores	1	2		1	

4.2 Results of comparison

Below, we show the results of the comparison of Program 2's current outcomes to the needed outcomes. An orange-filled cell indicates a match between a current and a needed learning outcomes, or a "1".

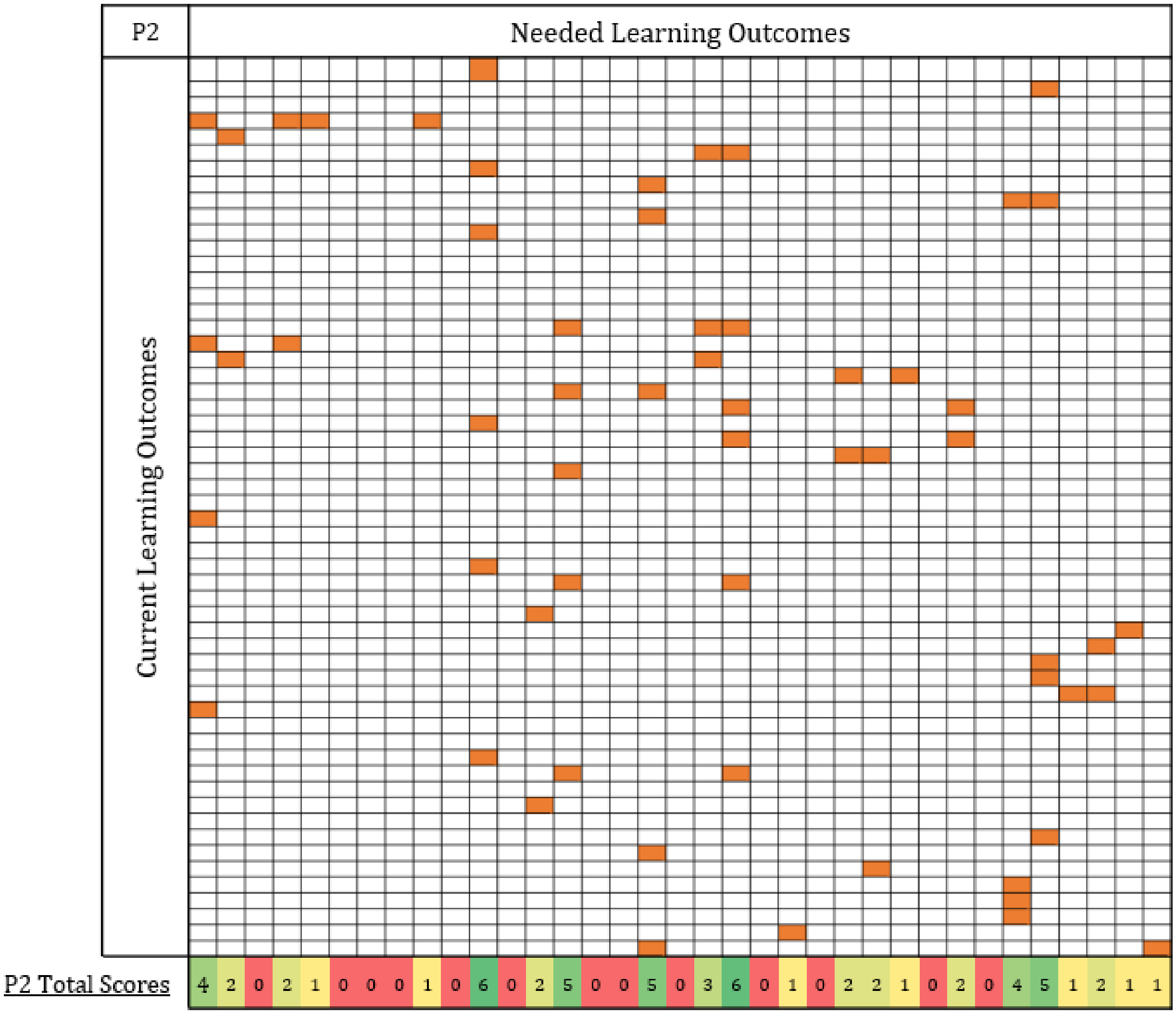
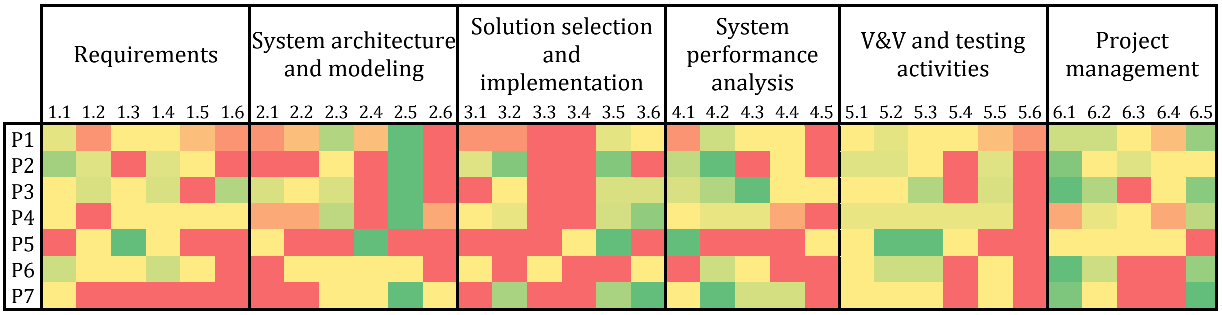


Figure 4.1 Comparison results for Program 2

As we mentioned earlier, we are interested in the total score a program obtained, per needed outcome. That number represents how often the outcome was mentioned in the syllabi of the courses required by the program and helps us determine to what extent we are teaching systems engineering students what they need to learn.

In Table 4.3, we summarize the results of the comparison of all identified undergraduate systems engineering programs in the US. Each row corresponds to a program and shows how often that program covered a certain needed learning outcome.

Table 4.3 Comparison scores per needed outcome and per program



4.3 Analysis of comparison results

4.3.1 By needed outcome

Table 4.3 shows that programs did not all cover the same set of needed outcomes, and taught students what they need to know to different extents. We first analyze Table 4.3 by column, and make the following observations:

- No needed outcome was covered by all seven programs.
- Eight needed outcomes were taught at least once and to any extent in six programs:
  - 1.1 Identify stakeholder needs
  - 1.4 Identify requirements at system level
  - 2.3 Develop high-level system architecture
  - 2.5 Model system
  - 3.5 Apply decision making methods
  - 4.2 Assess system effectiveness and performance
  - 5.5 Evaluate V&V test results
  - 6.5 Estimate cost
- Two outcomes were taught (to any extent) by only two programs:
  - 2.6 Identify interactions and dependencies between subsystems
  - 4.5 Analyze system resilience
- Three outcomes were taught (to any extent) by only one program:

- 3.3 Select optimal concept
- 3.4 Manage systems configurations
- 5.6 Document test plans, protocols and results

- Eleven outcomes were covered to the same extent across programs. Their average scores are represented in Figure 4.2.
    - 3.3 Select optimal concept
    - 3.4 Manage systems configurations
    - 4.5 Manage systems resilience
    - 5.6 Document test plans, protocols and results
    - 5.1 Perform system performance validation
    - 6.3 Write technical reports
    - 5.3 Define and develop test plans
    - 5.2 Conduct verification strategies on system…
    - 6.6 Estimate Cost
    - 2.5 Model System

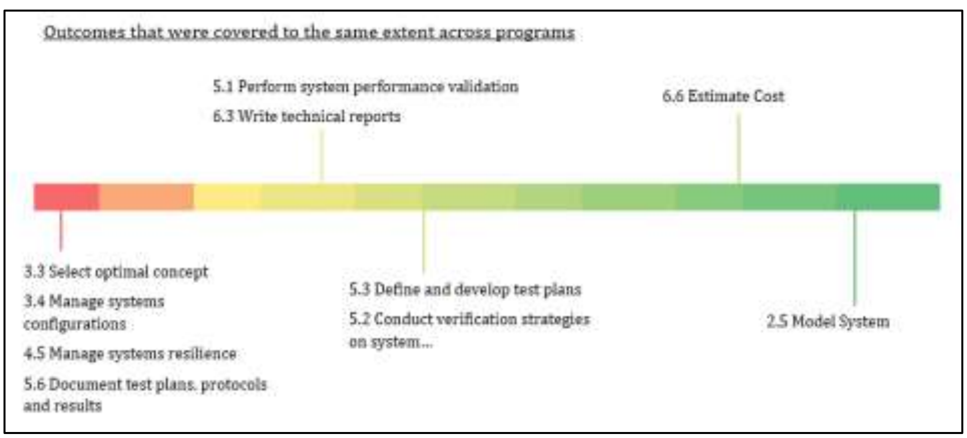


Figure 4.2 Average score of needed outcomes covered similarly across programs

4.3.2 By program

Here, we analyze Table 4.3 by row, to see whether each program emphasized particular sets of outcomes. We note that programs covered between 19 (Programs 5 and 7) and 30 (Program 1) needed outcomes each.

- Program 1 covered more needed outcomes than all other programs but focused the least on outcomes concerned with solution selection and implementation and with system architecture and modeling.
- Program 2 covered outcomes related to managerial tasks the most, and outcomes related to system modeling and architecture the least.
- Program 3 covered at least four outcomes from every category except Category 3, Solution Selection and Implementation.
- Program 4 covered at least four outcomes from each category.
- Program 5 mostly covered needed outcomes related to V&V activities, managerial tasks and requirements management.
- Program 6 focused on requirements and on V&V activities, but also emphasized some outcomes corresponding to managerial tasks.
- Program 7 focused on solution selection and on V&V activities, but also emphasized some outcomes corresponding to managerial tasks.

4.3.3 Further observations

- Cost analysis, although one of the least mentioned outcomes in our sources, is one of the most covered outcomes in programs.
- Five out of the seven programs we considered required that students learn about Human Factors; even though our six sources did not refer to human factors being part of the systems engineer's scope.
- of the systems engineer's scope.

4.4 Summary

In this chapter, we assessed each systems engineering program, by comparing its current learning outcomes (identified in Chapter 3) to our set of industry-based needed learning outcomes (generated in Chapter 2). We found a set of needed outcomes that almost all undergraduate systems engineering programs covered at least once, another set of outcomes that were covered by most programs to the same extent and a set of outcomes that were only covered by one or two programs. We also found that programs might be placing emphasis on outcomes that our sources did not consider to be that important.

5. CONCLUSION

5.1 Summary

The goal of this research is to determine whether we are preparing systems engineering students to meet the needs of the industry, by teaching them the necessary learning outcomes.

In Chapter 2, we determined the broad needs of the industry in terms of systems engineering. We captured these needs by looking at six different sources, each offering a different perspective. We then translated these needs into a set of “needed learning outcomes”. We found some evidence suggesting different worldviews on the systems engineering tasks. Our list of outcomes might not be 100% exhaustive and completely representative of the entire industry, but it is still valid.

In Chapter 3, we identified US universities offering undergraduate degrees in Systems Engineering. We looked at their required courses and extracted each program’s “current learning outcomes”, as mentioned in syllabi and courses description.

In Chapter 4, we compared the two set of learning outcomes, to determine whether programs are teaching systems engineering students what they need to learn. We found, for example, that eight of the 35 needed outcomes were taught at least once in six out of the seven programs.

Our approach was not based on a single definition of systems engineering, but on a variety of sources describing systems engineering from different perspectives. We did not interpret definitions of systems engineering or use them to set program objectives and then derive learning outcomes. We translated the needs of the industry directly into course-level learning outcomes. We also addressed previously identified gaps in the efforts to advance systems engineering education, by considering “lessons learned” in practice and compiled in research. We evaluated the state of systems engineering education by looking at programs in terms of what they teach only. We also identified areas needing further investigation, such as the existence of different worldviews on the systems engineer’s duties. We also found evidence suggesting that undergraduate Systems Engineering programs are preparing students to meet some but not all the needs of the industry.

5.2 Limitations and future work

5.2.1 What we say we teach VS what we are teaching

In Section 0, we explained how what we say we teach students can be different from what we actually teach them. In our research, we only considered what syllabi authors say they teach. Identifying what we are actually teaching students in a course might require the involvement of the course's instructor. Future work can look into contacting academic programs directly to obtain more details on what instructors actually teach.

5.2.2 What we are trying to teach VS What students are learning

Students do not always learn what we try to teach them. Our learning outcomes can be used to design a systems engineering survey and assess the gap between what we try to teach students and what they learn. By checking which systems courses that a student has taken, the learning outcomes they were taught can be identified. A student's performance on the survey can then indicate how well they learned the outcomes we tried to teach them, and if we are doing a good job teaching students what they need to learn.

5.2.3 Does the industry view the duties of a systems engineers differently?

In Chapter 2, we identified the needs of the industry, as described by six sources. In Section 2.4, we found that the six sources do not view the systems engineer's duties similarly. Some sources excluded duties mentioned in others. Checking whether different worldviews on the system engineer's duties exist was not within the scope of this research, but it would be interesting to do so, as future work.

A survey, similar to the one deployed by INCOSE Fellows, can be designed to identify worldviews on the systems engineering duties. The survey could, for instance, ask practicing and experienced systems engineers to rate the importance of certain tasks, or to indicate how often they do them, and to suggest additional ones that are not listed. The results of the survey, if they reveal low agreement levels and some divergence, could further motivate research into the worldviews of the systems engineer's duties.

5.2.4 How to assess *graduate* systems engineering programs?

Most Systems Engineering programs listed in the INCOSE and SERC Worldwide Directory of Systems Engineering and Industrial Engineering Programs, are graduate programs. The number of universities offering Systems Engineering as a graduate major is much higher, and a lot of the literature considers SE graduate programs more effective than an SE undergraduate education. But how do we assess the effectiveness of this type of programs? In section 3.1, we explained how our assessment approach only works on undergraduate programs, due to two things. One, graduate programs are usually very flexible and allow students to customize their plan of study. They might not teach the same things to all students who enroll in them. Second, ABET does not require graduate-level courses to have specific learning outcomes, the way it does for undergraduate programs. It is therefore more challenging to pinpoint what a graduate program teaches, especially in terms of learning outcomes. The answer to Research Question 2, "what are we currently teaching SE students?", might not consist of a set of extracted current learning outcomes, and the assessment method will therefore need to be different. Future work can investigate methods to assess graduate systems engineering programs effectiveness. The method can be designed to consider that not all instructors choose to list specific learning outcomes in their syllabi, and that not all schools allow public access to their courses' syllabi. It should also consider the large number of graduate programs that will need to be assessed.

5.2.5 Writing pre-requisite for our needed outcomes

Some outcomes we generated correspond to high levels of Bloom's taxonomy. In this research, we do not check how students can reach these levels, and do not identify the pre-requisite topics they need to learn in order to learn our needed outcomes.

APPENDIX A. INCOSE SYSTEMS ENGINEERING ACTIVITIES AND EXTRACTED EXPRESSIONS

In Section 2.1.1, we described the process that allowed us to extract four expressions from the first systems engineering activity, as defined by INCOSE (Systems and SE Definitions, n.d.). Here, we show the expressions we obtained after repeating that process on all ten activities.

- establish stakeholders' success criteria and concerns
- defining actual or anticipated customer needs and required functionality
- revising customer needs and required functionality
- investigating the solution space
- proposing alternative solution
- proposing operational concepts
- weighing value of solution and operational concepts (viability, utility, benefit at cost)
- selecting optimal concept
- architecting solution or set of solutions
- considering concepts of employment and usage
- modeling solution at each phase
- evaluating solution at each phase
- considering normal and exceptional scenarios, and an appropriate diversity of viewpoints
- establish required capability and performance
- increase confidence that the solution will work as expected and required
- avoiding or minimizing undesirable unintended consequences
- ensure solution is resilient
- ensure solution can evolve and adapt to changes in needs and environments
- provide prediction and assessment of system effectiveness and value
- define and manage interface, within system and between system and the rest of the world
- establish process and life cycle models
- consider complexity, uncertainty, change and variety
- implementing system management and governance processes for development and through life use and disposal
- proceeding with detailed design synthesis
- processing with integration
- proceed with solution verification and validation (ensuring the solution is fit for the intended purpose)
- consider aspects of dependability such as safety, security, reliability, availability, logistic support, and disaster recovery

- considering all necessary enabling systems and services, and end-of-life processes (transition to a replacement system, recycling of the retired one, nuclear decommissioning and waste disposal…)
- providing SE knowledge and information required by stakeholder
- ensure coherence of whole endeavor
- vision statement
- operational concepts
- business drivers,
- analyses and recommendations for decision support and the business case,
- architecture definition
- organization policies and processes
- required properties and interfaces
- ensure interoperability
- verification and validation criteria
- analysis and interpretation of test
- evaluation of test results
- anticipated operational usage
- appropriate system configurations for different scenarios
- supporting transition to use
- considering all aspects including people, process, information and technology
- periodically re-evaluating risks and opportunities
- periodically re-evaluating system effectiveness and value
- recommend corrective, mitigation, recovery actions
- maintenance and repair activities
- upgrade activities
- obsolescence management
- assessing information quality and integrity
- instituting metrics and incentives

APPENDIX B. SELECTED RESEARCH PAPERS AND THEIR MOST USED EXPRESSIONS AND WORDS

In Table 5.1, we list the Best Papers from 2018 and 2019 as selected by INCOSE, and we show the frequently used words and expressions that we used to build our needed learning outcomes.

Table 5.1 INCOSE Best Papers and extracted expressions

Paper title	Extracted words, bigrams and trigrams	Number of occurrences in abstract
Browning, 2018: Building models of product development processes: an integrative approach to managing organizational knowledge	Process	6
	PD	4
	Modeling	4
	IPM	3
Szajnfarber and Vrolijk, 2018: A facilitated expert-based approach to architecting "openable" complex systems	Open innovation	4
	Systems	4
	Crowd	3
	Problems	3
	Openable	3
	System	3
Issad et al., 2018: Scenario-oriented reverse engineering of complex railway system specifications	Methodology	4
	System	3
	Modeling	3
Raz et al., 2018: System architecting and design space characterization	System	11
	Design	11
	Architecture	6
	Decisions	5
	Design space	5
	Design space characterization	3
	Interactions	3
	System architecture	3
Broniatowski, 2018: Building the tower without climbing it: progress in engineering systems	Systems	6
	Engineering systems	5
	Systems engineering	3
	Field	4
	ES movement	3

Table 5.1 continued

Paper title	Extracted words, bigrams and trigrams	Number of occurrences in abstract
Ahn et al., 2018: Entropy-based system assessment metric for determining architecture's robustness to different stakeholder perspective	Different	8
	Perspective(s)	7
	System	6
	Stakeholders	4
	Robustness	4
	Proposed metric	4
	System architecture	3
	Configurations	3
	Perspective based decompositions	2
Togwe et al., 2018: Using a systems engineering framework for additive manufacturing	Life cycle	3
	Additive manufacturing	6
	Project crashing	2
Gallud and Selva, 2018: Agent-based simulation framework and consensus algorithm for observing systems with adaptive modularity	systems	5
	Framework	3
	Vehicles	3
	Coalitions	3
Delicado et al., 2018: Conceptualization of a T-shaped engineering competency model in collaborative organizational settings: problem and status in the Spanish aircraft industry	Team	4
	Engineering	4
	Assessment	3
	Competency(ies)	5
	T shaped competency model	2
Salado and Kannan, 2018: A mathematical model of verification strategies	Verification	5
	Verification strategies	3
	Mathematical model	3
	Systems engineering	3
	Theory of systems	2
	Practice	3
	Case	3
Rapp et al., 2018: Product development resilience through set-based design	Design	9
	Development	5
	Change	4
	Time	4
	Approach	4
	System	3
	Cost	3
	Changes	3
	Set	3
	Alternative designs	2

Table 5.1 continued

Paper title	Extracted words, bigrams and trigrams	Number of occurrences in abstract
Nolan et al., 2019: How Many Systems Engineers Does It Take To Change a Light Bulb?	Judgement	11
	Systems Engineering	5
	Crowd	4
Hecht, 2019: Quantitative Resiliency Analysis of Microgrids	Quantitative	3
	Resiliency	3
	Measures	3
	Performance and effectiveness	2
	Measures of performance	2
Sillitto et al., 2018: What do we mean by “system”? – System Beliefs and Worldviews in the INCOSE Community	Different situations	2
Biggs et al., 2018: Integrating Safety and Reliability Analysis into MBSE: overview of the new proposed OMG standard	Safety	4
	Reliability	4
	Aspects	3
	SysML	3
	MBSE	3
	Reliability analysis	2
Biggs et al., 2019: OMG standard for integrating safety and reliability analysis into MBSE: Concepts and applications	SysML	3
	Safety and reliability	2
	MBSE (Model Based Systems Engineering)	2
Watson et al., 2019: Appreciative Methods Applied to the Assessment of Complex Systems	Systems	10
	Complexity	8
	Characteristics	6
	Complex systems	4
	Appreciative inquiry	2
McDermott, 2019: Emerging Education Challenges for Resilient Cyber Physical Systems	Engineering	8
	Systems	8
	Education	7
	CPS (Cyber Physical Systems)	6
	Security	4
	Resilient	4
	Computing	3
	Engineering Education	2
	Systems engineering	2

Table 5.1 continued

Paper title	Extracted words, bigrams and trigrams	Number of occurrences in abstract
Buede et al., 2019: Innovation in the Spirit of Design Thinking	Process	4
	Iterative process	2
	Creation/Create	4
	Value	2
Bonnet et al., 2019: Augmenting requirements with models to improve the articulation between system engineering levels and optimize V&V practices	Engineering	3
	MBSE	2
	Model	4
	Verification and Validation	2
	Requirement	2

APPENDIX C. EXPRESSIONS EXTRACTED FROM NASA SE HANDBOOK

As mentioned in Section 2.1.3, we referred to Chapter Two of the NASA Systems Engineering Handbook to determine how NASA defines their SE Engine, and what additional managerial responsibilities their systems engineers are in charge of. Below, we show the expressions extracted from that chapter. The expressions cover the processes which make up the SE Engine (from the first section of Chapter Two) as well as the managerial responsibilities which do not fall in that engine (from the introduction of Chapter Two).

- safe and balanced design in face of opposing interests and constraints
- identifying and focusing efforts on assessments
- optimize overall design
- validating that goals of operational system
- when and where to probe
- system technically fulfills defined needs and requirements
- directs, communicates, monitors, coordinates tasks
- reviews and evaluates technical aspects
- development of concept of operations (ConOps)
- development of systems architecture
- defining boundaries
- defining and allocating requirements
- evaluating design tradeoffs
- balancing technical risk between systems
- defining and assessing interfaces
- oversight of V&V activities
- documenting technical plans, requirements and specification documents, V&V documents, certification packages
- supporting program and project planning and control
- accurate and timely cost and schedule information for the technical activities
- looking at big picture
- ensuring requirements, operational goals and stakeholder expectations are met
- technical characteristics of decisions including technical cost and schedule
- define and baseline stakeholder expectations
- generate and baseline technical requirements
- decompose requirements into logical and behavioral models
- convert technical requirements into a design solution that will satisfy stakeholder expectations

- top of structure to the bottom
- create design solution for each product
- verify
- validate
- transition up to next hierarchical level
- design solutions
- life cycle phase
- establish and evolve technical plans for project
- manage communication across interfaces
- assess progress against plans and requirements
- control technical execution of the project
- aid in decision-making process
- break down initializing concepts of the system
- integrate smallest product into greater and larger systems
- stakeholder expectations definition
- technical requirements definition
- logical decomposition
- design solution definition
- product implementation
- product integration
- product verification
- product validation
- product transition
- technical planning
- requirements management
- interface management
- technical risk management
- configuration management
- technical data management
- technical assessment
- decision analysis

APPENDIX D. EXTRACTED RESPONSIBILITIES FROM SELECTED JOB POSTINGS

Here, we list the responsibilities of an entry level systems engineer as described by job postings at Boeing, Lockheed, NASA JPL, Raytheon, Endotronix, Apple, Leidos and Booz Allen Hamilton.

- communication skills
- plan, design, schedule, and verify a balanced System Safety life-cycle solution
- Define the problem space
- generating system level concepts
- support concept selection
- Development of architectural concepts that meet top-level requirements
- Flow-down top-level requirements to sub-system areas
- Work with and integrate across sub-systems to define optimised system-level architectures
- external discussions with partners and customers
- requirements management
- determining, managing & flowing system level requirements
- understanding requirements from different viewpoints and impact of change.
- Experience in the architecture and design of systems using Systems Engineering principals, including the use of Robust Design techniques to achieve optimum solutions for the system of interest.
- Familiar with digital tools for design, analysis and automation of design processes, e.g. MBSE (Model-Based Systems Engineering)
- written and inter-personal skills.
- lead in identifying project requirements
- team environment
- work collaboratively across multi-disciplinary functions
- ensure that the system and components meet their product and marketing requirements
- Generate requirement specifications
- Write functional test plans and protocols
- Execute design verification and design validation testing
- Write clear and concise test reports
- Generate system level and component level risk documents (design FMEAs, use FMEAs, and hazard analysis)
- Evaluate design changes for impact on verification activities
- Evaluate industry standards to determine required testing and compliance (e.g. IEC 60601, ISO 14708)
- Assist in troubleshooting and problem solving for field issues
- Verifying and validating designs in a multi-disciplinary technical field
- writing requirements and technical reports

- written and verbal communication skills
- system architecture
- system design
- system integration
- technical management
- help to provide requirements analysis
- measure, track, and brief technical and programmatic risks and schedules
- preparing and presenting systems assurance reviews,
- identify requirements and deficiencies in hardware and software products,
- working in a team environment
- using requirements gathering and tracking
- Possession of oral and written communication skills
- development and integration of Commercial, Civil, and Military aerospace systems
- development and evaluation of system and subsystem level requirements
- development and documentation of test plans and procedures
- supplier interface and management
- coordination among a multi-disciplinary team
- experience effectively communicating within a team environment and have experience mentoring & guiding team members.
- system integration
- managing a team of engineers and scientists
- Perform system and vehicle level trade studies to integrate cutting-edge commercial solutions while balancing risk, cost, and performance
- Create workflows for the assembly, integration, and test of each satellite subsystem
- risk and failure analysis in system design
- point of contact for communications
- Experience with environmental testing
- Knowledge of common spacecraft failure modes
- Experience with a full satellite product life cycle
- formulation of mission concepts
- Mission Concept Development
- Flight Systems Integration & Test
- Concurrent/Collaborative Engineering
- Systems Modeling & Analysis
- Flight System Architecting and Behaviors
- Management and Development of Computer Aided Engineering Tools and Data
- Fault Management
- Project Verification & Validation
- systems-level analysis and modeling
- Defining systems architecture

- Defining requirements
- Defining use case analysis
- Defining tests
- documenting systems architecture, requirements, use case analysis, and tests
- Creating subsystem specifications by analyzing / modeling / characterizing components
- Defining subsystem and system level tests to characterize and/or guarantee parametric specifications
- Designing and performing laboratory experiments
- Managing project deliverables and schedules to meet commitments to product development teams
- Documenting the results of laboratory experiments
- performing system requirements derivation
- requirements management and system design development
- system integration
- system requirements verification
- system performance validation
- investigating laboratory anomalies
- analyzing system requirements and system designs
- developing test plan strategies
- development of test procedures and test execution
- engineering analysis of performance parameters as well as creating the necessary artifacts
- conducting reviews on each artifact
- configuration management of artifacts and products
- frequent communication with team members, customer stakeholders and with external team's Subject Matter Experts
- Applies systems engineering principles throughout the systems life cycle phases: Concept, Development, Production, Utilization, Support, and Retirement
- Communicates with other program personnel, government overseers, and senior executives.
- Stakeholder Requirements Definition
- Requirements Analysis
- Architectural Design
- Implementation, Integration, Verification, Transition, Validation, Operation, Maintenance, Disposal
- Project Planning
- Project Assessment and Control
- Decision Management
- Risk Management
- Configuration Management
- Information Management, and Measurement
- Project Portfolio Management

- Lifecycle Model Management

APPENDIX E. EXPRESSIONS EXTRACTED FROM INTERVIEWS

In Section 2.1.5, we mentioned how the interviews conducted by Aloisio (2016) with practicing systems engineers help us identify some "lessons learned" in systems engineering. Below, we list the expressions we extracted from the responses and which we used to generate our set of needed learning outcomes.

- fine line between defining aspects at the aircraft level and the subsystem level
- do not understand what the scope of their job
- clarity of communication;
- trying to communicate an idea unambiguously
- different environment in industry because clarity of communication is required
- trouble with understanding the big picture
- Process and requirements work is tedious and anyone will struggle with it
- lack knowledge and experience across different disciplines of design
- have to know to ask the right questions
- need to know what the designers consider, like environment, interfaces, signals, and structural loads
- understanding of aircraft, including their intricacies, coordination, and interfaces is needed
- not always familiar with the processes that interconnect and tie together (i.e. risk management, requirements management).
- lack of experience in the system as a whole
- different subsystems and subsequently how they tie together
- do not have the product experience.
- do not understand the interrelationship and functionality of the parts
- what they're looking at when they see a requirement.
- writing requirements
- do not recognize [requirements] are poorly designed.
- don't get exposed to requirement writing
- writing discipline
- interpretation of the requirements
- Being able to work as part of a large team
- system attributes and how every system works
- shoehorn in what the customer wants and make it work
- use the topdown approach.
- New hires should demonstrate flexibility and adaptability.
- accommodate a variety of expectations and demands.
- job for a generalist, not a specialist.
- work in and interact with a team

- open mind because some people focus on a single component
- look at the system from an overall perspective
- how aspects interact with each other at a high level
- work with other people, organize them, recognize potential problems and get people to work together
- managerial role to the subject matter experts and also be supportive to the chief engineers and program managers
- ensure deliverables to customers within specific time frames.
- experience in requirements
- Working well with a team
- leadership role to pull together thing like design reviews
- relate various pieces of a project together
- ability to multi-task.
- know how everything fits together
- team-oriented

REFERENCES

Aloisio, D. (2018). Lessons from Systems Engineering Failures: Determining Why Systems Fail, the State of Systems Engineering Education, and Building an Evidence-Based Network to Help Systems Engineers Identify and Fix Problems on Complex Projects. PhD Dissertation.

Armstrong, J. R., & Wade, J. (2015). Development of Systems Engineering Expertise. *Procedia Computer Science*, 689-698.

Arnold, E., Jackson, S., & Sillitto, H. (2017). Analysis of Results for the Systems Engineering Worldviews Survey. *28th annual INCOSE International Symposium.* Washington, DC.

Criteria for Accrediting Engineering Programs, 2019 – 2020. (2019). Retrieved from ABET: https://www.abet.org/accreditation/accreditation-criteria/criteria-for-accrediting-engineering-programs-2019-2020/

Goodlass, S. (2004). Can Systems Engineering be taught at Undergraduate Level? *Fourteenth Annual International Symposium of the INCOSE*, (pp. 945-955).

Jackson, S. (2018, August). *INCOSE.* Retrieved from INCOSE: https://www.incose.org/incose-member-resources/chapters-groups/ChapterSites/los-angeles/library-and-resources

Kossiakoff, A., Sweet, W., Seymour, S., & Biemer, S. (2011). *Systems Engineering Principles and Practice.* Wiley Series in Systems Engineering.

Lineberger, R., & Hussain, A. (2016). *Program management in aerospace and defense, still late and over budget.* 2016: Deloitte Development LLC.

Mar, B. (1997). Back to basics again -- a scientific definition of systems engineering. *INCOSE International Symposium.* Seattle.

NASA. (2020). *National Aeronautics and Space Administration.* Retrieved from NASA Systems Engineering Handbook: https://www.nasa.gov/connect/ebooks/nasa-systems-engineering-handbook/

Page, A. (1992). *Systems Engineering* . Wiley Series in Systems Engineering.

Systems and SE Definitions. (n.d.). Retrieved from INCOSE - International Council on Systems Engineering: https://www.incose.org/about-systems-engineering/system-and-se-definition