Abstract

This paper discusses a recent engineering graduate's viewpoint(s) about his career with United States Air Force. He discusses the importance of undergraduate engineering disciplines being taught with respect to systems engineering guidelines. He discusses how systems engineering should also be taught from a practical viewpoint. If the systems engineering process is added to the traditional engineering curriculum, it should highlight war-fighter's or customer's needs, their specifications, standards, and requirements. The paper stresses the idea that requirements engineering methodologies, a subset of systems engineering, should be taught. This paper discusses the use of requirements engineering to derive software and hardware requirements for avionics subsystems. It defines definitions and applications of systems engineering methodologies to develop a technological product that succeeds from one milestone decision to another and from one life cycle phase to the next. Recommendations are made on how to better prepare pre-graduate engineers for such a workplace.

Introduction

As a college student pursuing a bachelor's in electrical engineering, the author of this paper often heard terms like “Once you graduate, you won’t even use half of this stuff on your job”, or “I’ve been on two internships and still don’t know why we need to know all of this stuff”, or, “We don’t really need to know anything about the design… the technician handles all of that”, and finally, “I wish I had learned all this back when I was in college”! After entering the workforce, the author of this paper is able to empathize, to a certain extent, with people who made such comments. However, the comments highlight a gap between what is taught in many engineering schools and what is required in a real-world environment. Consequently, the author argues that an entrepreneurial, intrapreneurial, and systems engineering approach to teaching engineering should be adopted in undergraduate engineering curricula throughout the country based on a practical standpoint. Such an approach would go beyond the basic mathematical analysis and engineering management curricula currently taught at many institutions in America. This approach would focus on customer needs, for instance the project life-cycle and requirements engineering, using, specific examples from the United States Air Force’s (USAF’s) or Department of Defense’s (DoD’s) Acquisition Life-Cycle references and documentation. Thus, an overview of systems engineering will be discussed, along with requirements engineering, systems development, the USAF or DoD acquisition life cycle, and requirements as they relate to the acquisition life cycle.

Intrapreneurship and Entrepreneurship

Since graduation, this author has learned that intrapreneurship and entrepreneurship are commonly practiced. Intrapreneurship can be broadly defined as the application of entrepreneurial capabilities to the development of new ventures within an existing firm. For the practice of intrapreneurship it is irrelevant whether the firm is private or public. Intrapreneurship is different from the practice of entrepreneurship.

Entrepreneurship involves the activity of organizing, managing, and assuming the risks of a private enterprise [1]. Hence it is a private enterprise that might have transitioned from intrapreneurship to entrepreneurship phase or it was private from the beginning. Systems engineering is used for both intrapreneurship and entrepreneurship. From a Department of Defense (DoD) and United States Air Force (USAF) perspective, this author’s employer, its systems engineers, practice intrapreneurship, where they help develop companies, sometimes called Systems Program Offices (SPOs), within the government framework.
**Systems Engineering**

Some academic institutions view the systems engineering process as being similar to the systems integration role. The bias is toward the technology’s design, application, and installation. For DoD or USAF, and other enterprises the role of the system engineer is much broader. The bias is toward not only the system’s technological development but also program management and other war-fighting and business related matters. USAF’s systems engineering activities involve the processes, resources, costs, and program management concerning the design of an entire product for the customer, or in the case of the USAF or DoD, the war fighter [2]. During undergraduate education, the author was taught according to mathematical descriptions that described an input into a box representative of electronics processing functions; the electronics functions within the box were representative of filtering, signal conditioning, amplification, and conversion processes that might be performed on such inputted signals; and the output from the box was representative of the modified or the improved inputted signals with, for stability purposes, a feedback to the input for the box.

This traditional engineering methodology taught the author how an electronics system might work. However, it and the engineering curriculum, in general, placed almost no emphasis on the actual development of the electronics system itself, how the design effort was funded, or how to use the systems engineering process to understand how such an electronics function sub-system might be integrated with other sub-systems to form a system.

For example, customers request that aircraft have specific and unique performance requirements, the vehicle be delivered according to schedule and at the lowest cost possible. As a result, the responsible systems engineers determine the types of skill sets required for the technical management activities and the processes involved in making the developing system a reality. The systems engineers determine how to integrate and install new components onboard an airplane, the risk involved in doing such, and configuration control or configuration management (see Figure 1) [3].

![Integrated Risk Management](image)

**Figure 1. Integrated Risk Management [3]**

As future trends in systems engineering move toward increasing complexity, an increased need for analytical and software tools are needed [2]. Consequently, it seems undergraduate course work in engineering should include a required course dedicated to the basic understanding of these tools for the students regardless of discipline.

Traditionally, engineering professors focus on component level development curricula.

Although component level development is important, private and public sector engineers tend to focus more on aggregated components and less on the individual components. It appears what is considered to be traditional engineering curricula at a university is actually non-traditional and what is
treated as non-traditional at a university seems to be traditional for engineering practitioners. Increasingly, more of the component level systems engineering development is being outsourced [4]. If this argument is true, it seems necessary that engineering curricula needs to include systems engineering as part of the mainstream curricula as opposed to the converse.

The author of this paper has observed that it takes considerable amount of time for junior engineers on both the system and subsystem levels to become proficient at performing systems engineering, which is due, in part, to minimum knowledge of the fundamentals. The exceptions being: engineering graduates who while students worked with systems engineers as part of their internships or co-operative education programs. However, the author advocates systems engineering being taught as a part of mainstream engineering curriculum at our nation’s engineering schools, and that internships and co-operative education programs be used as a means for reinforcement of the learning and development of those skills.

**Engineering Requirements within the Systems Engineering Process**

Any system developed for a customer must meet requirements to be acceptable. The USAF is one such customer. For example, when the Air Force acquires a sub-system, system, or air vehicle the process involves capital and costs; different types of contracts; a myriad of DoD and USAF directives, guidelines, and instructions; and systems engineering processes along with concerns about safety-of-flight, ethics, and risk management [3]. Thus, requirements for each system and sub-subsystem section must be decomposed from the functional requirement(s) to the physical, including the description of object(s) and state condition(s) relating back to the desired function(s). The desired functions are driven by software, firmware, and logic. Those are skill sets required of the software systems engineer. Software systems engineering, for instance, involves the overlap of various systems. These points of convergence must be understood in order to meet the requirements of the given system [5]. Consequently, an engineering curriculum should present Requirements Engineering (RE) from a perspective that involves multiple customers, safety, and many other components (see Figure 2)

![Figure 2. Requirements Engineering Activity Cycle](image-url)
How can these diverse topics be integrated in such a way as to be realistic, reasonably cost effective, and duplicable? The author suggests a basic course in systems development.

**Systems Development**

The junior engineers this author has become acquainted with have experienced difficulty understanding the process by which a system is acquired. Consequently, it is difficult to understand decomposition from a system to the function needed by each group within the systems engineering community. It is also difficult to develop the mindset needed to encourage engineers to care what other groups need. Emphasizing the subject of concept development within engineering schools is a great way to counter the negative consequences mentioned above. As a result, the junior engineer will be encouraged to work cooperatively with other engineers, technicians, and all other people involved in their respective projects. Since processes for concept development vary depending upon the system, requirements, and customers involved, specific examples from industry are suggested as case studies. For the following example, the concept of aircraft systems development within the United States Air Force will be discussed.

The Air Combat Command or war-fighters determine if there is a need. The war fighters make requests to System Program Offices (known as the SPOs), Groups or Wings determine if a new or existing aircraft or air vehicle or avionics system need to be developed or modified to give them a desired capability. As a result, an avionics system for an aircraft may be decided upon [3]. Eventually the requirements reach the systems engineering team, a system or device is chosen, and the location of that component within the plane is finally decided upon. Next, the mechanical, aerospace, and electrical design groups are given their task concerning electrical power analysis, safety-of-flight and other parameters. The electrical section must create the interface documentation for power, data, and radio frequency devices. Then, the engineers within this section must analyze the electrical loads, safety, and risk of Electromagnetic Interference/ Electromagnetic Compatibility (EMI/EMC) based on the interface material. This sub-process is in addition to similar processes conducted by the other groups mentioned above. After approval by the appropriate personnel, the avionics system is ready to be built and/or installed on an existing aircraft.

When a junior engineer is given the position of “lead engineer” on such a project, he or she may not set a realistic schedule or develop a realistic cost estimate without being trained to grasp a basic understanding of the system life cycle. In general, the lifecycle of a system includes the requirements, alternatives, design, implementation, system tests, operation, maintenance, and retirement/disposal of the system [6]. The fist step in developing a given system is to state the problem, which includes a function the system must do or discrepancy that must be eliminated. This step, coupled with constant communication with the customer(s), will lead to a clear definition of system requirements [6]. The requirements determining processes includes the minimal conditions the system must have in order to be accepted by the customer, as well as what would further please the customer [6]. Once these steps are done, requirements can be validated, alternatives investigated, models and simulations verified, and designs completed. There are many details within this systems life cycle description. Thus, a brief overview of the acquisition life cycle phases within the Department of Defense will be described for the purpose of clarity.

**The Acquisition Life Cycle**

The acquisition process is best described by dividing it into phases defined by milestones, which describe an important decision point in time that pre-determines the start of the next phase. Milestones are also known as decision reviews [7]. A milestone must contain quantifiable information that can be validated in a prescribed check list, which is agreed upon by the customers [7]. For example, the refinement of the technology for a new avionics communication system can be described as milestone A [8]. The actual development and demonstration phase, which includes modeling and simulation, is described as milestone B. Milestone C can be described as the production phase, in which the first systems are produced. Finally, milestone D will include the
operations and support tasks needed to support a given system throughout its life-cycle, ending only after the system undergoes its disposal process, following the end of its useful life-cycle [8].

There are numerous processes within these milestones to ensure that not only the proper system is developed, but the proper procedure for system development and design is followed. For example, during Milestone B, a Preliminary Design Review (PDR) is conducted. During this process, the customer reviews the provider’s proposed plans for designs and testing, based upon the basic needs of the customer [8]. Also, the hardware, software, test equipment, and other supportive components are reviewed in descending order of system to unit. When approved, the provider is ready to move forward with detailed design that involves the hardware and software of the given system. Next, is the Critical Design Review (CDR). During this process, the customer reviews and approves “build-to” and “code-to” specifications. In addition, all hardware, software, test equipment, and tooling should be reviewed in ascending order from unit to system [8]. After the two processes mentioned above, a system is generally still in need of long-term support. Milestone C includes the fielding of designed systems for operational testing and evaluation of a given system [7]. Once a system performs to the satisfaction of the customer, the other aspects concerning supportability, maintenance, and the quantity of systems to be fielded by the producer are additional challenges faced by the systems engineers. This highlights milestone D. Also, such cases re-enforce the role requirements engineering play in the overall life-cycle of a system.

**Requirements Engineering in Relation to the Project Life Cycle**

Now that the importance of the systems engineering life-cycle has been covered, the basics of requirements engineering must be examined in further detail. For example, an avionics software system may need periodic upgrades, or be amendable to changes in technology related to hardware/software interfacing. Also, when a hardware device is added to an airplane, wire gauges for the consumption of power must be determined. Thus, a process for configuration control is very important [3]. In either case, a management process must be in place to control changes to requirements that will eventually be needed. Figure 3 is an example of configuration control [9].

![Figure 3. Configuration Control Example [9]](image)

Other examples to consider could include an electrical power plant, or programmable devices. Even after system design and installation, the changing needs of a mission or technology will affect the useful life of the system, thus prompting system design teams to plan and estimate the useful life of the given system (see Figure 4) [9]. Based on this fact, a suggested approach to teaching the basics of requirements engineering would be to use examples similar to the above mentioned as case studies in an introductory class.

9.B.5-5
The requirements concerning risks will need to be made consistently throughout the systems design process. And in the case of the Department of Defense, the analysis of risk management continues throughout the project life cycle [3]. As a consequence, the systems architects or systems architect team must consider alternative designs throughout the creation and testing of the system (see Figure 5) [10]. Such a process also requires team members in all groups to consider safety, cost, ethics, etc., without placing too much importance on just one of these parameters. Many organizations have conflict concerning how flexible the requirements can be [10]. A suggested approach to teaching risk management as one of the requirements would be to discuss priorities for requirements engineering. A specific example can be a cost versus safety discussion within various class room examples.

Figure 4. Suggested Class Discussion Example: Air Traffic Control System [9]

Conclusion

The author is well aware of the rigorous course load already part of engineering disciplines within American Colleges. However, re-examination of the engineering curricula can be used to determine priorities, add classes, or subtract classes, which may reveal how engineering education gaps between what graduates need to understand versus what the schools presently teach can be lessened. The author suggests that the concepts described in the paper be gradually introduced, starting in the sophomore year of the student’s curriculum. Since
the prerequisites to engineering disciplines require
an in-depth study of calculus based mathematics
combined with the physical sciences, the author
also suggest re-examination of the type of
“homework” and testing given in junior level and
senior level classes. Redundancies that go beyond
the basic reinforcement of mathematics and science
principals may be safely eliminated to make room
for additional systems oriented study. In addition,
textbooks dealing with systems engineering could
be structured for sophomore through senior level
engineering disciplines. Although some educators
can argue that internships should serve in this
capacity for students, one must realize that
internships are generally specific to the respective
company. Also, the competitive nature of
internships ensures that many students may not get
the “right” internship needed to provide a decent
education in the systems engineering process.
Thus, if the engineering curricula is changed to
provide a foundation in the basics of system
engineering, internships would be an added
motivation to future engineers who are eager to
effectively. Consequently, the next generation of
engineers will be passionate about their
contributions to society, which in turn can create
improvements in any industry.

Acknowledgements

Special Thanks to the following people:

Robert A. Crist, 46 Test Wing, TSIS, Eglin AFB
John C. Perry, 46 Test Wing, TSI, Eglin AFB
Job Gammons, 796 CES/CEOCD, Eglin AFB
Robert L. Riley Jr., AFRL/MNAV, Eglin AFB
James T. Mitchell, 796 CES/CEOCD, Eglin
A FB
LT. Sonia Betchloff, 796 CES/CEOOM, Eglin
A FB
Keith W. Jones, Ph.D., ASC/XRS, Wright-
Patterson AFB

References

[1] Editorial, September 2000,”Intrapreneurship:
Spinning Off a New Company”, INC.com,

Volume: 15, Issue: 10, Jubilee Issue, IEEE
9-10.

ASC/EN”, ASC/EN, Wright-Paterson Air Force
Base.
[https://www.en.wpafb.af.mil/...engineering.asp]

Cites Outsourcing As Wages Drop and U.S. Loses
221,000 Jobs” Volume: 53, Number: 8, Electronic
Design, Penton Media, pp. 25.

[5] Sommerville, Ian, (January/February 2005),
“Integrated Requirements Engineering: A Tutorial”,

Systems Engineering? A Consensus of Senior
Engineers”, Tucson Arizona, Systems and
Industrial Engineering, University of Arizona.
http://www.sie.arizona.edu/sysengr/index.html

Technology, and Logistics Life Cycle Management
Framework”, 2004, Defense Acquisition
University, Fort Belvoir, Virginia

[8] Tutorial “ Integrated Defense Acquisition,
Technology, and Logistics Life Cycle Management
Framework Flowchart”, Defense Acquisition
University, Fort Belvoir, Virginia

System Engineering: A Tutorial”, California State
University, Sacramento, California, IEEE Computer
Society.(Vol. 35, No. 4) pp. 68-73.

[10] Luman, Ronald, R, Richard S. Scotti, Fall
1996, “The System Architect Role In Acquisition
Program Integrated Product Teams”, Acquisition

24th Digital Avionics Systems Conference
October 30, 2005