ASEE's 123rd Annual • Conference & Exposition • New Orleans, LA • June 26-29, 2016

Paper ID #15385

Systems Engineering and Capstone Projects

Dr. Fred J. Looft, Worcester Polytechnic Institute

Prof. Looft earned his B..S, M.S. and Ph.D. degrees in Electrical Engineering at the University of Michigan. After a brief period on industry, he joined the faculty of WPI 1n 1980 where he is now a professor in the ECE department and a founder of, and Academic Head of the Systems Engineering program. His interests include projects based education, curriculum development, international study abroad programs and mentoring, and autonomous robotic systems..

Systems Engineering and Capstone Projects

Abstract

Systems Engineering (SE) methods are increasingly being integrated into capstone design projects as a critical component of capstone design competitions, through mentoring during capstone project advising, and through capstone course syllabi development. In this paper, we describe an "engineering science" course developed specifically to teach selected SE topics and designed to primarily prepare third and fourth year undergraduate students for their engineering (ABET) capstone project. The course was developed using an inverted classroom format where students view short, topic-specific videos and read selected key topic papers and tutorials prior to class meetings. Class time is then used to work on topic specific problems in an engaging team based setting. This paper presents the motivation for the course, the course structure and syllabus, and recommendations based on how to improve the course based on student input and first time offering observations.

Introduction

According to the International Council on Systems Engineering[17] (INCOSE):

> "*a system is a construct or collection of different elements that together produce results not obtainable by the elements alone*"

while systems engineering (SE) is:

> "*an engineering discipline whose responsibilities is creating and executing an interdisciplinary process to ensure that the customer and stakeholder's needs are satisfied in a high quality, trustworthy, cost efficient and schedule compliant manner throughout a systems entire life cycle*"

A more complete review of what people have in mind what they discuss the meaning of systems engineering is provided by Fraser and Gosavi[18] but for the purpose of this paper, we will focus on the key points that:

- Systems engineering is an interdisciplinary, well defined and described process incorporating as series of steps that enhance the likelihood of developing a successful system.

SE in the Undergraduate Curriculum

Based on the above very basic definition and understanding of SE methods and importance, there are fundamentally three reasons identified in the literature to include SE principles in the undergraduate engineering curriculum:

- in support of ABET student outcomes[1],
- to improve and inform capstone project development and design methods[7], and
- to address industry demand for SE knowledge[8-10]

From an ABET perspective, Criterion 3 STUDENT OUTCOMES states that ABET accredited engineering programs should demonstrate that program graduates have:

(c) an ability to design a system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability

(e) an ability to identify, formulate, and solve engineering problems,

(k) an ability to use the techniques, skills, and modern engineering tools necessary for engineering practice.

Outcome "c" can be interpreted in multiple ways, but is perhaps the most self-evident in support of including SE competencies when compared to the (excerpted) definition of systems engineering from INCOSE[13]:

> Systems Engineering is an interdisciplinary approach and means to enable the realization of successful systems. It focuses on defining customer needs and required functionality early in the development cycle, documenting requirements, then proceeding with design synthesis and system validation while considering the complete problem.

Similarly, one interpretation of outcome "e" is that this outcome supports the inclusion of SE methods when one considers that capstone projects and SE methods overlap in areas such as thoroughly understanding the needs of an engineering design problem, correctly and competently communicating with stakeholders, and fully documenting requirements that need to be satisfied for an engineering design problem[6,7]. Outcome "k" can relate to the importance of understanding that SE methods are considered a critical aspect of "techniques, skills and modern engineering tools" needed for the engineering practice and thus also supports industry demand.

The inclusion of SE methods specifically in capstone courses, projects and other engineering undergraduate offerings has been reported at numerous ASEE conferences and various publications and has been a topic of growing interest for university programs for perhaps the last 15 years [e.g. 2, 5, 7]. Without detailing an exhaustive review of the literature, there are several notable recent papers and workshop activities that provide rationales and guidance for SE methods inclusion. One particularly interesting example of SE inclusion, albeit for first year project students, is documented by Parnell and Kwinn[3] who describe the value of, and how SE methods are taught starting in the first year of studies at their university. The approach they describe is rooted in a first year course taken primarily by SE program majors, but also taken by many students outside the SE program as a gateway course into other, primarily engineering programs. A unique aspect of their entry level SE course is that they developed a book[4] as a companion to the course development effort.

Others have described courses, modules and other activities that address teaching SE methods during the third or fourth year of study and applied specifically to capstone projects. Developments have included:

- course sequences that start with a course of study specifically tailored toward interdisciplinary capstone projects but can lead to the MS degree in SE[7],
- capstone SE training modules[5], and
- reports on how SE methods serve as a foundation for the development of systems developed for national engineering design competitions[2].

What each of these papers, and many others describe are predominantly one-time implementations of SE competency syllabi for a specific SE program, capstone course or competition.

More recently, workshops and development efforts have focused on two key topics:

- defining SE core-competencies for capstone projects, and
- defining the types of educational material (e.g. modules, videos, tutorials) that would support the inclusion of SE methods into *any* capstone project/course/program.

The two most recent endeavors in these areas were presented and summarized during the 2015 ASEE Annual Conference and covered the topics of defining and integrating SE competencies[8] and integrating systems engineering into engineering education[9,10], with a focus on both defining competencies and specific materials for capstone projects. Simoni et al.[8] detail and explain desirable system competencies that capstone project students (among other) should learn. These include:

1. Applying a system stakeholder view of values, trade-offs and optimization of a system. Stated another way, is the system concept, design and operation as it evolves what the stakeholders really want?
2. Defining a project as interconnected subsystems.
3. Understanding a system's interactions and states (modes).
4. Specifying system technical requirements.
5. Creating and analyzing high-level designs including concept architectures and implementations, and (for example) HW/SW functional trade-offs.
6. Assessing solution feasibility, completeness and consistency.
7. Performing failure mode and risk analyses.

Contrasting these competencies are those of Squires et al. [9,10] which were discussed and developed as part of an INCOSE Academic Forum meeting and summarized at the ASEE Annual meeting. Final competencies identified included:

1. Systems Science and Fundamentals
    a. Understanding Complexity
2. Systems Thinking
    a. Interdependencies
    b. Problem Analysis
    c. Total Life-Cycle View
    d. Multiple and Holistic Perspectives
3. Design and Analysis
    a. Systems Architecture and Analysis
    b. Requirements Analysis
    c. System Modeling
    d. Trade off and Decision Analysis
4. Technical and Project Management

a. Dealing with Uncertainties and Change

On the surface, these two lists would seem to be dissimilar. However, one[8] was developed more from an applied, engineering program perspective while the other[9,10] was developed from a "system" perspective of the holistic engineer and, when presented at 2015 ASEE[10], included a break-down of topics that could relatively easily be matched to those of Simoni et al. In other words, the lists were not significantly dissimilar when the details of each area of competency were compared.

<u>Core SE Competencies for Capstone Courses</u>

Capstone project courses do not provide sufficient time to adequately present and practice a wide range of relevant SE methods and material. As a result, if one seeks to incorporate SE competencies into a capstone course/program, one needs to start by considering the framework of the course/program and then identify the core SE concepts and competencies that will provide the most "SE value added". As noted above, guidance for the selection of essential SE topics comes from sources such as the INCOSE and ASEE workshops focused on SE competency models, curriculum inclusion and material development needs [6,7,12].

For the course described herein, a subset of topics was developed based on previously published workshop and academic forum reports generated from previously reported SE course and module development experiences, industry expert opinions, ASEE SE competency papers and, to some extent, student design capstone project problem areas based on more than 35 years of project advising by the author in consultation with his SE colleagues. However, prior to describing the specific concepts selected for our course, we will first describe the unique capstone projects based environment at our university to put the course development in context.

Capstone Projects at WPI

Projects are at the heart of the WPI curriculum, requiring students to apply the knowledge learned in classes and labs to real-world problems. At WPI there are *two* capstone projects required of *all* undergraduate students. One of the projects is unique to WPI, is typically completed in the *third year* of study, and is known as the Interactive Qualifying Project (IQP). This project is specifically designed to focus on the relationship between technology and society. As noted in the IQP handbook[14] (excerpted):

- The IQP is the means which WPI has chosen to make science and engineering students aware of the role of their professions in society. The importance of such an understanding has been reinforced by the ABET Engineering Criteria 2000, which require that engineering programs demonstrate that their graduates have "the broad education necessary to understand the impact of engineering solutions in a global societal context."
- The IQP is by design interdisciplinary. Students obtain practice in dealing with unstructured, open-ended, interdisciplinary problems, opportunities to work independently with peers and extensive experience in writing about previously unfamiliar concepts utilizing new terminology.

The second project is completed in the fourth year of study and for engineering majors satisfies the ABET requirement for "*a major design experience based on the knowledge and skills acquired in earlier course work and incorporating appropriate engineering standards and multiple realistic constraints*" [1]. This project, known as the Major Qualifying Project (MQP), represents a three course equivalent and is normally a full-year commitment by a team of students working with a faculty project advisor. As shown in **Figure 1** below, what is different about the WPI capstone project is that it is not classroom based, but instead is advised by a WPI faculty member in the student's major (1B). By comparison, at most universities, the capstone is taught within t he context of a classroom environment (1A).

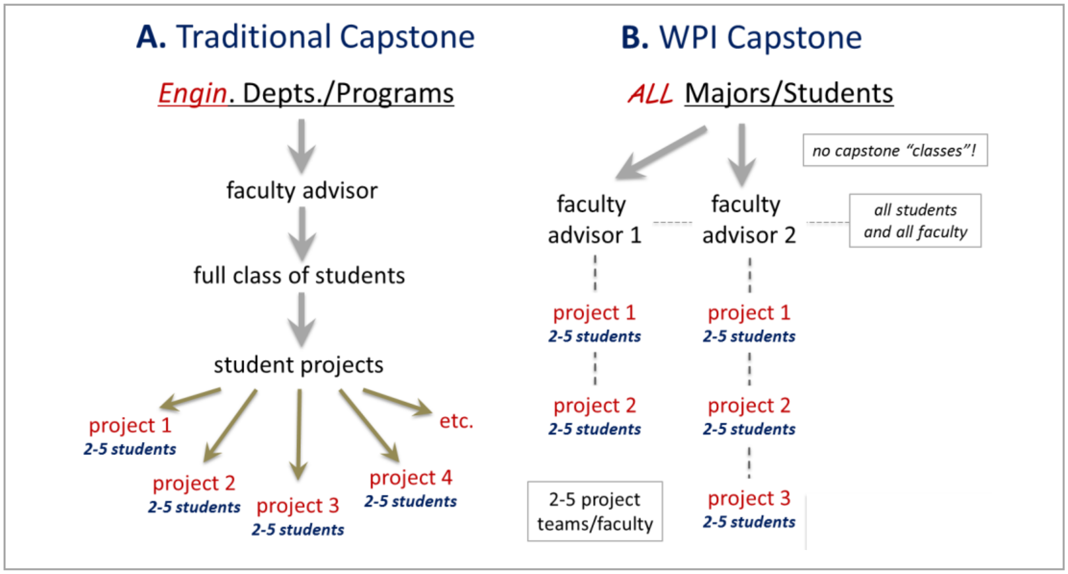

Figure 1. More Traditional and WPI Approach to Capstone Project Advising.

Course Format, Competencies, Outcomes

There were two distinct steps in the development of our SE FOR CAPSTONE PROJECTS course. The first was to teach an *experimental* (independent study) version of our proposed SE FOR CAPSTONE PROJECTS course to a small group of third and fourth year capstone project students to discern how best to integrate selected topics and in-class exercises into individual team projects, and to gain some direct student feedback on the relevance of the topics covered. The second step was to use the lessons learned from the experimental course offering to develop a fully approved, ES version of our SE FOR CAPSTONE PROJECTS course targeted at third and four year undergraduate students.

Relative to the first step, an undergraduate independent study course covering core SE concepts and competencies applied to capstone projects was offered in the first quarter of 2014 (12 students enrolled). Nearly all of the students who participated in the initial experimental offering of the core SE concepts and competency class were starting their fourth year capstone project. A few, however, were working on their third year society/technology project. The projects ranged from an autonomous robot development capstone to various computer/electrical engineering capstone projects, and one third year IQP focused on redeveloping a first year engineering projects based mechanical engineering design course. Unlike the formal follow-on course which was taught in an inverted classroom format, the experimental offering was taught in a traditional small class format: about half of each period covered lecture material, while the other half was spent engaging students to solve a specific problem relative to the topic being discussed.

Specific lessons learned from this experimental course offering included the following.

- Teach the class using an inverted format
  - the students clearly preferred that more class time be used for in-class, engaging topic development and problem solving
- Use ill-defined engagement problems - in other words, problems that are not well bounded, where the system need is not necessarily well characterized, where the decision basis for different designs are not fully specified, and where in general the specifications for a system leave significant room for creative thinking and design!
  - interestingly, using ill-defined problems was seen as an advantage to teaching "systems thinking" concepts, and helping students think through the essence of a problem or goal was advantageous for generating in-class discussions
- Structure all homework problems to be applicable to each student's (or team's) own capstone project (whether an IQP or MQP)
  - homework was SE methods based - making it easy to assign a common homework problem to many different project teams - for example, "write two use cases that …"
  - the homework resulted in material that could be used by a student or team in the final capstone project report later in the year
- Modeling and simulation, which was not covered in the experimental offering, needed to be included
  - the students were surprisingly naïve about the value the modeling and simulation as a decision making component of any SE method
- Include software (SW) SE as a critical competency topic
  - most students did not have a strong background in SW systems as part of their own program of study, yet nearly every student project had a software development component
- Continue the communications intensive aspects of the course
  - WPI students are not required to take a specific writing course as part of their distribution requirements. However, in addition to satisfying a general humanities distribution requirement where most student learn some professional writing and presentation skills, the IQP and MQP described earlier serve as project based learning environments for applied communications skills development. By making this SE course writing intensive, and focusing the homework assignments on various sections of their final capstone reports (e.g. problem statement, stakeholder identification, risk assessment and management, etc.), the students develop increased proficiency in their written and oral presentation skills.

Subsequent to the experimental course offering, a regular WPI *Engineering Science* (see Discussion) course, ES 3501, A PROJECT-BASED INTRODUCTION TO SYSTEMS ENGINEERING was created and approved by WPI faculty. The course description for ES 3501 is as follows.

> Systems Engineering is a multifaceted discipline, involving human, organizational, and various technical variables that work together to create complex systems. This course is an introduction and overview of the methods and disciplines that systems engineers use to define and develop systems, with a particular focus on capstone projects. The course

will include specific integrated examples, projects, and team building exercises to aid in understanding and appreciating fundamental principles. Topics covered will include: Introduction to Systems Engineering; Requirements Development; Functional Analysis; System Design; Integration, Verification and Validation; Trade Studies and Metrics; Modeling and Simulation; Risk Management; and Technical Planning and Management.

The course outcomes were specified as follows.

At the completion of this course, students will be able to:

1. Explain what a system is, what systems engineering (SE) is and what is meant by the SE development process.
2. Explain the classical SE Vee diagram, and be able to elaborate on different phases of system development activities along different points of the diagram.
3. Explain what a good requirement is and provide examples of good requirements.
4. Explain what is meant by validation and verification. Provide some examples of the application or methods appropriate to each.
5. Explain how to conduct a trade study, the value of trade studies, and different metrics that can be used as a basis for a trade study.
6. Explain the value of modeling in the design of a system. Give an example of a system development effort that would benefit from modeling.
7. Explain what technical performance measures are and how they are used in the development of an engineering system.

Course Text

No single text is designated for this course. Students are instead directed to several excellent on-line references that have been used very successfully and include the following.

FHWA: http://ops.fhwa.dot.gov/publications/seitsguide/index.htm and http://www.fhwa.dot.gov/cadiv/segb/views/index.cfm

The web site is comprehensive, has both text and graphic illustrations with interactive web material, includes templates for reports and processes where appropriate, and is appropriate for both learning and teaching SE principles. An example of a link through the FHWA web site that provides excellent tutorial material is here: http://ntl.bts.gov/lib/jpodocs/repts_te/13621.html.

FDOT: http://www.dot.state.fl.us/trafficoperations/ITS/Projects_Deploy/SEMP.shtm

The Florida Department of Transportation (FDOT) web site has many good tutorial slides covering the SE life cycle.

DoD: http://www.acq.osd.mil/se/index.html

The material on this web site provides some interesting counterpoints to the FHWA and FDOT websites, but is often too detailed for general capstone project use. An example of DoD web material that is useful however are the Architecture Framework descriptions found here: http://dodcio.defense.gov/Library/DoDArchitecture Framework/dodaf20_viewpoints.aspx. While it would not be expected for students on a capstone to develop a full suite of DoDAF views, the link above provides guidance to students for specific system views that could be asked for by course, project or even corporate mentors for a specific capstone project.

SEBOK: http://sebokwiki.org/wiki/Guide_to_the_Systems_Engineering_Body_of_Knowledge_(SEBoK)

The SE body of knowledge web site is useful for case studies, general SE methods and process overviews and definitions, and general "systems" information.

Syllabus/Schedule/Format

It was intended (and accomplished) that the course would be offered in an inverted classroom format to maximize the time available for in-class engagement and problem solving during normal course meeting times. Specifically, using an inverted classroom presentation form, students are directed to on-line web learning modules and reading material for each class topic. Class time is then used to apply the module material to specific problems that relate nuanced interpretations and understanding in an engaging environment.

Table 1 details the major topics covered in this course. With regard to this table, it is important to note that the WPI academic schedule is based on students taking four terms in an academic year, with a fifth (optional) summer term. Each term is seven weeks long and three full time courses/term constitute a full (undergraduate) load. The SE ES capstone course is offered in the first term of the year and it is expected that students will take this course concurrent with starting their capstone project (MQP but could also be the IQP). A one-term course typically meets four times/week and courses that have laboratory sessions (not this course) also layer in a weekly lab section. A simple calculation shows that as a result of this course/term structure, a typical class will meet 28 times/term. The total number of entries in the table is 26 since two are set aside for a mid-term and a final exam.

Table 1. ES 3501 Syllabus Topics

Topic	# Class Periods	Topics Covered
What is a system?	1	introduction, overview, course format, grading, etc.
What is Systems Engineering?	1	INCOSE definition, SE roles, management structures, education required, certification, jobs, etc.
The SE life cycle	1	Vee diagram, concept of designing a system from idea, to design, to test, to deployment and eventually retirement
Stakeholders	2	types, (proxy or direct, passive or active etc.), priority, role, etc.
Needs Analysis	2	identifying/solicitation, concepts of good needs, prioritizing, key needs, traceability to stakeholders, documenting, etc.
Validation and Verification	2	what is meant by *verification* and *validation*, what is validation and how is it perform, what is verification and how is it performed, documenting V&V, when in life cycle, etc.
CONOPS	2	definition, CONOPS format/documentation, gap analysis, use cases, discovering requirements, unknowns, context diagrams, etc.
Requirements	2	"good" requirements and types, incorporating standards, traceability (validation) and testing (verification), documenting, managing, *not* gold plating, key requirements, etc.
Technical Measurements	1	MOPs, MOEs, KPPs, testability, etc.
Ilities	1	definitions, categories, different types of ilities, etc.
System Design	2	alternatives, considerations for study and evaluation, partitioning, requirements flow down, functional partitioning and flow down, etc.
Trade Studies	1	what, why, how, methods (e.g. excel, probabilistic, regression, etc.), etc.
Technical Design Reviews	1	why, how, what, when, SE role & how to conduct, phase gates, etc.
Risk Identification and Management	2	identification, prob. vs. severity, considerations for safety and appropriate ilities, management, TEAM, etc.
Test, Design for Test	2	when/were in life cycle, how, what, documentation, management, etc.
Software SE	3	software life cycles, development methods, scrum[11], etc.

Each class period follows essentially the same topic presentation process outside of class, and the same approximate in class format/engagement exercise schedule:

- prior to the next class meeting
  - students view one or two short video tutorials (~5-8 minutes each)
  - students read one or more tutorials on a specific topic
  - as appropriate, students review a "standard" for documentation, presentation or process for the specific topic
- during class
  - the topic material is reviewed and questions are answered (~5-10 minutes)
  - a small team engagement problem is presented (~5 minutes)
  - small teams (~3-4 students/team) work on the problem (~15 min.)

    - teams present their results and other teams/participants critique (~15 min.)
    - summary (~5 min)
- homework is then assigned to reinforce the topic

Assessment of Learning Outcomes

A key aspect of any course development effort is a plan to assess course outcomes. For this course, homework assignments and other course material are designed to directly measure learning outcomes. An example of a lesson plan that would support assessment of both the specific topic lecture material as well as general course learning outcomes is shown in **Figure 2**.

Learning Outcome

#3: Explain what a good requirement is and provide examples of good requirements.

Lesson Plan (Introduction to Requirements)

Prior to Class

- View/study Introduction to Requirements video tutorial
- Read/study FHWA web material on requirements: http://www.fhwa.dot.gov/cadiv/segb/views/document/sections/section3/3_5_1.cfm
- Download and read a short professional paper on requirements: e.g. http://www.mitre.org/sites/default/files/publications/se-guide-book-interactive.pdf (p314-318)

In Class

- Instructor: Answer questions about the material reviewed outside of class.
- Instructor: Present a problem for groups of students to work on that relates to the topic of "introduction to requirements"

  Example: Your team is tasked with designing a new e-bicycle. 1. Briefly state any system design assumptions that will impact your requirements. 2. Write at least two of each of the following types of requirement: functional, non-functional, legal/standards.
- Students: Break into teams of 3-4 students and work on the problem. Write your results on the white board. Present and explain why the requirements are "good". Respond to critiques from other class members.
- Instructor: Summarize what was learned from the requirements discussion. Clarify any misunderstandings, etc.
- Instructor: Review the homework assignment.

HW Assignment

For your own specific project (MQP or IQP), do the following.

1. **Write** two "good" requirements for your project for each of the following types.
   a. functional requirement
   b. non-functional requirement
   c. software requirement
   d. legal/compliance requirement
2. For each requirement, **explain** why the requirement is considered a good requirement based on accepted INCOSE/SEBOK definitions and terminology.
3. **Validate** each requirement (i.e. show traceability to one or more needs and one or more stakeholders - note: stakeholders and needs would have been identified in a previous HW assignment and in-class discussion).
4. Indicate which requirements are **key** requirements and explain why they are key.
5. Briefly explain how your team will **verify** (test) each requirement.

Lessons Learned, Changes Needed

Although the course enrollment was small (~14 students), a few student comments from both offerings (experimental, approved) are provided below (excerpted).

- "This course made me appreciate why Systems Engineering is a formal practice. My biggest takeaway is that SE is the distilled processes and considerations that long-time experts intuitively go through to make sure projects succeed, but formalized and generalized for the rest of us to use. I found myself having flashbacks to my Software Engineering class, because both involve dialogs between the needs the stakeholders (think they) have and the capabilities of the development team to meet the corresponding requirements. The power in both SE and SWE is that your team is perfectly justified in not doing something if it wasn't an explicit requirement, and can avoid that distraction and can identify things that really are core to the project, or things that would cause the project to fail if they weren't addressed. SE helps bring that kind of focus and consideration to a project, even though it involves some overhead."
- "I really enjoyed taking your course this term and I feel like it has helped guide our [capstone] project. This course can be applied to any project and has improved my writing. I actually did a stakeholder, needs, and gap analysis for my ME 5105 course (Renewable Energy) this term. The systems engineering approach helps organize my writing and I know where to focus my efforts at the start of every project. *I wish all of my MQP partners took this course.* One challenge my group will face will be getting the other group members who haven't taken this course to take the systems approach."
- "I really enjoyed taking this course. Companies seem to value systems engineers for their project management skills and big picture type thinking. I plan on pursuing my master's degree in ME and I will definitely consider taking some SE courses."
- "Our team had no idea how to approach our capstone design project until we took your course. We were scared, overwhelmed and totally lost. Now we feel like we understand the problem and how to make significant progress on it and we are excited!"

Homework, in-class exercises and course format feedback resulted in the following thoughts about how to improve the course for future offerings.

- Add more tutorial videos.
  - The first formal presentation of the course was supported by video tutorials for perhaps only 2/3 of the course. Additional tutorials will be developed to span the full set of topics covered by the course.
- Reserve more time for SW SE and Scrum.
  - Although all engineering students who are not CS majors do take at least a few programming courses, few take a course in SW engineering or SW SE. As a result, students do not have any background in understanding the differences between HW/ complex system and SW SE.
  - Nearly every interdisciplinary capstone project has a SW design component that would benefit from applying SW SE methods.
- Incorporate at least one lecture on Project Management (PM) and the differences and similarities between SE management and PM.
  - Students specifically asked for more information on this topic.
  - Consider an evening session add-on "short course" covering these and related topics.
- Continue to advertise that the course material is applicable to IQP capstones (specific to WPI, but applicable to non-engineering capstones at other universities as well).

- Consider adding a separate evening tutorial on "systems thinking" applied to technology ⇔ society capstone projects.

Discussion, Summary and Conclusions

What is clear to us after reviewing the literature and conference proceedings prior to developing our course, and from what we learned during and after developing our course, was that:

- There are many ways to teach SE principles to capstone project students.
- The structure of a capstone "course" is highly dependent on the specific structure of capstone projects course/advising format at each university.
- The skill level of the faculty member delivering the capstone course SE material is highly variable. For faculty who are not experts or SE practitioners the availability of tutorial material can be particularly useful (Squires et al. [9,10]).
- There are no "best" SE concepts instructional method. Each approach must be tailored to a program's needs and delivery/advising architecture.
- The definitions of core SE competencies are dependent, in part, on the specific project focus and capstone course format. While perhaps two-thirds of the SE material could be considered "core" for all capstone projects, there is still the other one-third of the "core" material that could be adapted as needed to specific projects or development needs. For example, capstone projects that are SW intensive should spend more time on SW life cycles, use cases, storylines, Scrum and other SW specific SE tools, methods and implementation practices. Similarly, capstone projects that are process oriented might spend more time on systems thinking topics and, if particularly risky, more time on risk management (esp. identifying and quantifying risks, risk mitigation, etc.)

Why an Engineering Science Course?

At WPI Engineering Science (ES) courses are considered foundational to multiple disciplines and cover the breadth of common engineering topics such as Introduction to Engineering, Material Science, Thermodynamics, Controls Engineering, Environmental Impacts of Engineering Decisions, and Computer Aided Design to name a few. According to the 2014-15 Undergraduate Catalog [16] (*italics* added):

> In the formation of a program of study for any engineering or science student, *it is important to emphasize a significant number of interdisciplinary courses which form the fundamental building blocks of so many scientific and engineering activities.*
>
> In addition to those courses in science and mathematics which are an important part of every engineer's background at WPI, *there are a number of courses containing subject matter common to a variety of disciplinary interests.* These courses are known as the "engineering science group" and are often taught jointly by members of more than one department
>
> Every engineer, for example, needs to have some knowledge of graphics, the communications tool of engineering: of thermodynamics, the consideration of an important aspect of energy and its laws: of mechanics, solid and fluid, static and dynamic, the treatment of forces and their effects on producing motion. *These and certain other courses of either basic knowledge or broad application are grouped in the engineering science series to provide special focus on them for all students interested in applied science or engineering.*

Within the context of justification for ES courses noted above, the SE faculty believed that all engineering/science students would benefit from understanding the basic framework for designing systems as embodied in well-known SE principles. Since nearly all of the WPI engineering programs require that students take a select number of ES courses as part of their distribution requirements, no catalog or distribution requirement changes would be required to enable a student to take an SE Capstone methods course to satisfy distribution requirements if we placed our course in the ES domain.

Where to from here?

As pointed out by the respected author C. S. Wasson[12] (excerpted):

> "The overarching objective is for the engineering student to be capable of developing fully integrated mindset that instinctively links the subject matter of the topic examples listed below into a coherent solution with consistency throughout its documentation. The information content in each must be logically linked, concise, consistent and traceable within the overall system design solution framework."

For faculty considering incorporating SE core concepts, Wasson provides a list of SE subjects[12] that he considers core and which can serve as an excellent starting point subject to course/capstone needs, available time, etc. The work of Squires et al. and others[6,10,11] should also be referenced to review slightly different perspectives of what are considered SE core competencies.

From the perspective of the course described herein, and in particular with reference to the tools and material discussed as part of the INCOSE Academic Forum (AF) reports (summarized in 10), the following activities will be accomplished.

1. All existing course videos will be re-recorded to improve topic specificity and adaptability to other programs and courses.
2. Additional videos will be created to span the full set of topics covered by the course syllabus.
3. All videos will be made available on-line to all interested users.
4. Ibid - the course lecture plans, homework assignments, tutorials, etc.

Perhaps the best way to end this paper is to continue to quote two recommendations from Wasson[12] which also strongly support the INCOSE/AF work of Squires et al[10,11] and the development of the course described herein:

> "… two recommendations emerge to solve many of the performance problems and issues that plague system development projects today:
>
> - **Engineering Education** - Institute a Fundamentals of Systems Engineering course as a degree requirement for undergraduate engineering curricula that addresses concepts, principles, and practices. Specifically, a course focused on the SE concepts, principles, and practices of system development introduced prior to capstone project courses …,

- **Organizations -** Shift the organizational Specify-Design-Build-Test-Fix Paradigm to a scalable SE methodology-based education and continuous improvement paradigm that goes beyond SE philosophy and theory.

As a consequence, organizational competency will be enhanced, and project performance will more predictably correlate with organizational capability maturity results. "

References

1. ABET *Criteria for Accrediting Engineering Programs, 2015-16 Accreditation Cycle*, http://www.abet.org/wp-content/uploads/2015/05/E001-15-16-EAC-Criteria-03-10-15.pdf
2. Guerra, L. *Applying Systems Engineering to the Lunabotics Mining Competition Capstone Design Challenge*, 2013 ASEE Ann. Conf. ID: #6932
3. Parnell, G. S., Kwinn M. J. *The First Course* ASEE 2008 Zone I Professional Papers Proceedings, http://www.asee.org/papers-and-publications/papers/zone-proceedings/zone1/2008-professional
4. Parnell G. S., Driscoll P. J., Henderson D. L. (Editors), *Decision Making for Systems Engineering and Management*, Wiley Series in Systems Engineering, Sage A. P. Editor, Wiley and Sons Inc. (2008)
5. Pichette D. W., Ellis R. D., Bryzik W., Kim K-Y., Lai M-C. D., Kim, Y. S. *Integration of Systems Engineering Training Modules into Capstone Courses across College of Engineering Departments*, 2011 ASEE Ann. Conf. ID: #1270
6. Schindel, W. D., Peffers S. N., Hanson J. H., Jamell A., Kline, W. A. *All innovation is Innovation of Systems: An Integrated 3D model of Innovation Competencies*, 2011 ASEE Ann. Conf. ID: #1370
7. Sheppard K. G., Nastasi J., Hole E., Russell P. L., *SE Capstone: Implementing a Systems Engineering Framework for Multidisciplinary Capstone Design*, 2011 ASEE Ann. Conf. ID: #1206
8. Simoni, M., Andrijcic E., Kline, W. A. *Integrating Systems Competencies into the Curriculum of Any Engineering Discipline*, 2015 ASEE Ann. Conf. Workshop ID: U213C - expanded workshop report here: http://www.omgwiki.org/MBSE/doku.php?id=mbse:patterns:asee_2015_systems_ competencies_workshop_06.14.15
9. Squires A., Looft, F. J., Virani, S. *Integrating Systems Engineering into Engineering Education*, 2015 ASEE Ann. Conf. Workshop ID: U448
10. Squires, A., Looft, F. J., Virani, S., *Integrating Systems Education into Undergraduate Engineering Education plus INCOSE Spring Academic Forum Summary* Workshop Report, 2015 ASEE Ann. Conf. Workshop Report http://cdn2.hubspot.net/hubfs /357441/ASEE_2015_INCOSE_Workhop_Report.pdf
11. Sutherland, J. *Scrum: The Art of Doing Twice the Work in Half the Time*, Random House Business Books, ISBN: 9781847941107 (2014)
12. Wasson, C. S. *System Engineering Competency: The Missing Course in Engineering Education*, 2012 ASEE Ann. Conf. ID: #3389
13. INCOSE: What is Systems Engineering? http://www.incose.org/AboutSE/WhatIsSE
14. WPI: Project Based Learning. https://www.wpi.edu/academics/ugradstudies/project-learning.html
15. WPI: Undergraduate Catalog, IQP. http://www.wpi.edu/Images/CMS/IGS/IQP-Handbook-one-file.pdf *and* https://www.wpi.edu/academics/igsd/iqp.html
16. WPI: Engineering Science, Undergraduate Catalog, pages 73-74. http://www.wpi.edu/Images/CMS/Pubs-Catalogs-Ugrad/UGCat14-15FinalWEB.pdf
17. INCOSE: http://www.incose.org/AboutSE/WhatIsSE
18. Fraser, J. Gosavi, A. "What is Systems Engineering", ASEE Ann. Conference, AC 2010-267 (2010) available here: http://web.mst.edu/~gosavia/fraser_gosavi.pdf