Graduate Reference Curriculum for Systems Engineering (GRCSE), version 0.5

A product of the Body of Knowledge and Curriculum to Advance Systems Engineering (BKCASE™) project

GRCSE™ version 0.5 Release for Global Review December 2011

Copyright© 2011 Stevens Institute of Technology

This work is copyright 2011 by Stevens Institute of Technology. All rights reserved. Images used by permission remain the property of their respective copyright holders.

You may use the contents of this work on the conditions that (a) you acknowledge that your use is with permission of Stevens Institute of Technology, and (b) you indemnify and hold harmless Stevens Institute of Technology, the editors, and the authors, from any and all liability or damages to yourself or your hardware, software, systems, or third parties, including attorneys' fees, court costs, and other related costs and expenses, arising out of your use of this work, irrespective of the cause of said liability.

Stevens Institute of Technology makes this work available on an "as is" basis and makes no warranty, expressed or implied, as to its accuracy, capability, efficiency, merchantability, or functioning. In no event will Stevens Institute of Technology, the editors, or the authors be liable for any general, consequential, indirect, incidental, exemplary, or special damages, even if they have been advised of the possibility of such damages.

Preface

GRCSE is a part of the Body of Knowledge and Curriculum to Advance Systems Engineering (BKCASE™) project of the Systems Engineering Research Center (SERC). BKCASE is creating two products: GRCSE and the *Systems Engineering Body of Knowledge* (SEBoK). The SEBoK is a guide that provides overviews of key knowledge areas and topics, and directs users to references from the systems engineering literature. The SEBoK encompasses much of the content that is referenced in GRCSE. Version 0.5 of SEBoK was released in September 2011.

The BKCASE™ author team comprises invited experts from industry, government, academia, and various professional associations. These authors followed an iterative, evolutionary approach to create the two primary BKCASE™ products. The BKCASE™ author team met in workshops approximately every three months between December 2009 and October 2010, leading to the limited-review release of v0.25 of both GRCSE and SEBoK. GRCSE v0.25 was released in December 2010 to selected members of the SE community with the invitation to review and provide the necessary feedback to develop subsequent versions. This feedback was received and forms the basis of the current version, v0.5.. Version 1.0 is scheduled for release in late 2012.

Professional society participation in the creation of the GRCSE project is essential to ensure that it has the desired impact on global graduate education. Both INCOSE and the Institute of Electrical and Electronics Engineers (IEEE) Computer Society are providing representatives to serve as authors to the project. Discussions are underway with both societies with the expectation that the two societies will jointly take on the evolution and maintenance of GRCSE after the v1.0 documents are published.

Acknowledgments

Each BKCASE™ author, shown in Table 1 below, has supported the writing of the SEBoK, GRCSE™, or both. The 18 authors from 14 organizations who specifically supported GRCSE are shown in bold in the table. They came together selflessly and with the support of their organizations to improve global systems engineering education. Special thanks go to Tim Ferris, the lead GRCSE author.

We gratefully acknowledge the strong support of the U.S. Department of Defense (DoD), which provided financial support for this project to the Systems Engineering Research Center (SERC) as well as key members of the author team. We also acknowledge the support of our collaborating organizations, including the International Council on Systems Engineering (INCOSE), the Institute of Electrical and Electronics Engineers (IEEE) Computer Society, the National Defense Industrial Association (NDIA) Systems Engineering Division, and the Association for Computing Machinery (ACM). We especially thank the SERC, which has provided administrative and logistical support.

The team would also like to thank the technical editors, Abraham Raher and Justin Gercken, for their efforts to improve the readability and quality of GRCSE.

Table 1. Table of BKCASE authors. Those in bold contributed to writing GRCSE (table developed for BKCASE).

Author	Organization
Rick Adcock	**Cranfield University, UK**
James Anthony	**Sevatec, Inc., USA**
Erik Aslaksen	Sinclair Knight Merz, Australia
Richard Beasley	Rolls Royce, UK
Johan Bendz	Bendz & Co., Sweden
Barry Boehm	University of Southern California, USA
Stuart Booth	Office of the Secretary of Defense, USA
John Brackett	**University of Boston, USA**
Chuck Calvano	Naval Postgraduate School, USA
Aaron Eng Seng Chia	**Naval University of Singapore, Singapore**
Kyung-il Choe	Hankuk University of Foreign Studies, South Korea
Edmund Conrow	Management and Technology Associates, USA
Paul Croll	CSC, USA
Cihan Dagli	Missouri University of Science and Technology, USA
Judith Dahmann	MITRE, USA
Heidi Davidz	UTC Pratt and Whitney, USA
Jeremy Dick	Integrated Systems Engineering Ltd., UK
Charles Dickerson	Loughborough University, UK
Dov Dori	Massachusetts Institute of Technology, USA
Joseph J. Ekstrom	**Brigham Young University, USA**
Dick Fairley	Colorado Technical University, USA
Alain Faisandier	Association Francaise d'Ingenierie Systeme and MAP Systeme, France
Timothy Ferris – Lead Author	**University of South Australia, Australia**
Kevin Forsberg	Center for Systems Management, USA

Author	Organization
G. Richard Freeman	**Air Force Institute of Technology, USA**
Sanford Friedenthal	SAF Consulting
Brian Gallagher	Northrop Grumman, USA
Michael Henshaw	Loughborough University, UK
Thomas Hilburn	**Embry-Riddle Aeronautical University, USA**
Nicole Hutchison	**Stevens Institute of Technology, USA**
Peter Jackson	**Cornell University, USA**
Scott Jackson	University of Southern California, USA
Mo Jamshidi	University of Texas San Antonio, USA
Cheryl Jones	U.S. Army, USA
Chul Whan Kim	Korea National Defense University (KNDU), South Korea
Naohiko Kohtake	**Keio University, Japan**
Harold “Bud” Lawson	Lawson Konsult AB, Sweden
Yeaw lip “Alex” Lee	Defence Science and Technology Agency, Singapore
Ray Madachy	Naval Postgraduate School, USA
James Martin	Aerospace Corporation, USA
Greg Mayhew	Boeing, USA
Steven Mitchell	**Lockheed Martin, USA**
Kenneth Nidiffer	Software Engineering Institute, USA
David Olwell	**Naval Postgraduate School, USA**
Andy Pickard	Rolls-Royce, USA
Ricardo Pineda	University of Texas at El Paso, USA
Daniel Prun	**Ecole Nationale de l’Aviation Civile (ENAC), France**
Art Pyster	**Stevens Institute of Technology, USA**
Garry Roedler	Lockheed Martin, USA
Jean-Claude Roussel	European Aeronautical Defence and Space Company (EADS), France
Seiko Shirasaka	**Keio University, Japan**
Bob Shishko	Jet Propulsion Laboratory, NASA, USA
Hillary Sillitto	Thales Group, UK
John Snoderly	Defense Acquisition University, USA
Alice Squires	**Stevens Institute of Technology, USA**
Bill Stiffler	Raytheon, USA
Massood Towhidnejad	**Embry-Riddle Aeronautical University, USA**
Guilherme Travassos	**Guilherme Horta Travassos, Federal University of Rio de Janeiro (UFRJ), Brazil**
Ricardo Valerdi	Ricardo Valerdi, University of Arizona, USA
Mary VanLeer	**Mary VanLeer, Perceptive Systems, Inc., USA**
Qing Wang	Qing Wang, Institute of Software Chinese Academy of Sciences, China
Brian Wells	Brian Wells, Raytheon, USA
Mary Jane Willshire	Mary Jane Willshire, USA
Brian White	Brian White, CAU<SES, USA
Ken Zemrowski	Ken Zemrowski, TASC, USA

Art Pyster

Art Pyster
Stevens Institute of Technology
Editor-in-Chief and Project Leader

David Olwell
Naval Postgraduate School
Co-Editor-in-Chief and Co-Project Leader

James Anthony
Sevatec, Inc.
Editor

Stephanie Enck
Naval Postgraduate School
Editor

Nicole Hutchison
Stevens Institute of Technology
Editor

Alice Squires
Stevens Institute of Technology
Editor

Executive Summary

The Graduate Reference Curriculum for Systems Engineering (GRCSE™) is a set of recommendations for a systems-centric master's level graduate program in systems engineering, together with implementation guidance for a university to satisfy those recommendations. It is one of two products of a larger project, the Body of Knowledge and Curriculum to Advance Systems Engineering (BKCASE™). The other product, which preceded GRCSE and on which it draws, is the *Guide to the Systems Engineering Body of Knowledge* (SEBoK). (Pyster and Olwell (eds.) et al. 2011)

GRCSE describes a program for a professional master's degree focused on developing student ability to perform systems engineering tasks and roles.

GRCSE includes:

- a set of objectives describing the near-term career goals of a graduate who successfully completes a graduate program based on the curriculum;
- a set of outcomes to be met by a student to successfully complete a graduate program based on the curriculum;
- a set of student skills, knowledge, and experience assumed by the curriculum for the entering student, not intended as entrance requirements for a specific program, but as the starting point for the curriculum's outcomes;
- an architectural framework to communicate and support implementation of the curriculum;
- a description of the Core Body of Knowledge (CorBoK) taught in the curriculum to achieve the outcomes;
- guidance on implementation and assessment, and
- a set of use cases indicating how GRCSE will support the needs of various stakeholders using it for specific purposes.

Additional materials included in this document are:

- the fundamental philosophy for GRCSE development described in a set of guiding principles;
- a discussion of how GRCSE will evolve to remain effective; and
- a glossary, references, and other supporting material.

GRCSE is to be a living document that is revisited regularly and updated when necessary to ensure relevance to the rapidly evolving systems engineering discipline.

Summary of Objectives

Objectives are important statements about the future professional activities of the graduates of a program. They are used to set the program outcomes and to tailor the offerings of a program to support the career expectations of the student and employer communities. They are key differentiators between programs, and explain the diversity of program offerings.

GRCSE recommends that all programs set formal objectives to help define the top-level requirements for their programs and offers the following sample objectives.

1. *Effectively analyze, design, and implement feasible, suitable, effective, supportable, affordable, and integrated solutions throughout the life cycle of systems of systems, enterprises, services, and products. This could be tailored by explicitly stating the types of systems that graduates develop and a given domain (e.g., aerospace or telecommunications) or by specifying a portion of the system life cycle.*

2. *Successfully assume a variety of roles in multi-disciplinary teams of diverse membership, including technical expertise and leadership at various levels.*

3. *Demonstrate professionalism. Grow professionally through continued learning and involvement in professional activities. Contribute to the growth of the profession. Contribute to society through ethical and responsible behavior.*

4. *Communicate (read, write, speak, listen, and illustrate) effectively in oral, written, and newly developing modes and media, especially with stakeholders and colleagues.*

Summary of Student Outcomes

Outcomes are statements about the competencies possessed by a graduate upon completion of the program. Ideally, outcomes are derived from objectives. Most programs examined by the GRCSE authors currently have outcomes, either formally or informally. There is a wide diversity in these outcomes, which contributes to the confusion as to what constitutes a graduate degree in systems engineering.

Graduates of a master's program that aligns with the GRCSE recommendations will:

1. *Achieve designated Bloom's levels of attainment for each SEBoK topic contained within the CorBoK foundation.*[1, 2]

[1] GRCSE uses Bloom's taxonomy of educational outcomes to describe the levels of attainment expected at graduation. Appendix B provides a detailed discussion of Bloom's taxonomy and the rational for using Bloom's taxonomy in GRCSE.

[2] The foundation contains the knowledge that every graduate student in systems engineering should master, along with the level of mastery. The concentration currently has two branches, systems design and development, and systems engineering management. A student is expected to master the material in one of these two branches as well as the foundation. The foundation and concentrations together constitute the CorBoK.

2. *Achieve designated Bloom's levels of attainment for each SEBoK topic contained within one of the concentration focus areas, as appropriate for the type of master's program or for an individual student's interest.*

3. *Achieve a Bloom's Synthesis level of attainment for at least one topic from the CorBoK (either foundation or concentration).*

4. *Demonstrate the ability to perform SE activities in one application domain, such as defense, aerospace, finance, medical, transportation, or telecommunications.*

5. *Apply systems engineering principles to address one application type, such as safety-critical or embedded systems, or one property, such as security, agility, or affordability.*

6. *Comprehend and appreciate the challenges of applying systems engineering to realistic problems as part of a multi-disciplinary team.*

7. *Be an effective member of a multi-disciplinary team, effectively communicate both orally and in writing, and lead in one area of system development, such as project management, requirements analysis, architecture, construction, or quality assurance.*

8. *Be able to evaluate alternative system solution strategies, including how well different solutions relate to the identified problem, and express relevant criteria to ensure solutions are selected against a holistic systems perspective.*

9. *Be able to reconcile conflicting requirements by finding acceptable compromises within the limitations of cost, time, knowledge, risk, existing systems, and organizations.*

10. *Be able to learn new models, techniques, and technologies as they emerge, and appreciate the necessity of such continuing professional development.*

11. *Comprehend the relationships between systems engineering and other disciplines, such as project management, human factors, and other engineering fields as discussed in the SEBoK, and be able to articulate the value proposition of these disciplines for systems engineering.*

12. *Demonstrate an understanding and appreciation of software engineering necessary to develop current and future product, service, and enterprise systems.*

13. *Demonstrate knowledge of professional ethics and the application of professional ethics in decision-making and systems engineering practice.*

Summary of Expected Background

Graduate systems engineering programs exist to provide a broad common framework for students coming from a wide variety of undergraduate backgrounds. To accommodate that breadth, GRCSE presumes that an entering student has:

- *The equivalent of an undergraduate degree in engineering, the natural sciences[3], mathematics, or computer science.*
- *At least two years of practical experience in some aspect of systems engineering. This experience should include participation in teams and involvement in the life cycle of a system, subsystem, or system component.*
- *Demonstrated ability to effectively communicate technical information, both orally and in writing, in a program's language of instruction.*

Summary of the GRCSE Curriculum Architecture

The curriculum architecture is organized into six components: Preparatory Knowledge, Foundation Knowledge, Concentration Knowledge, Domain-Specific Knowledge, Program-Specific Knowledge, and a mandatory Capstone Experience. The relationship between these elements is portrayed in Figure 1.

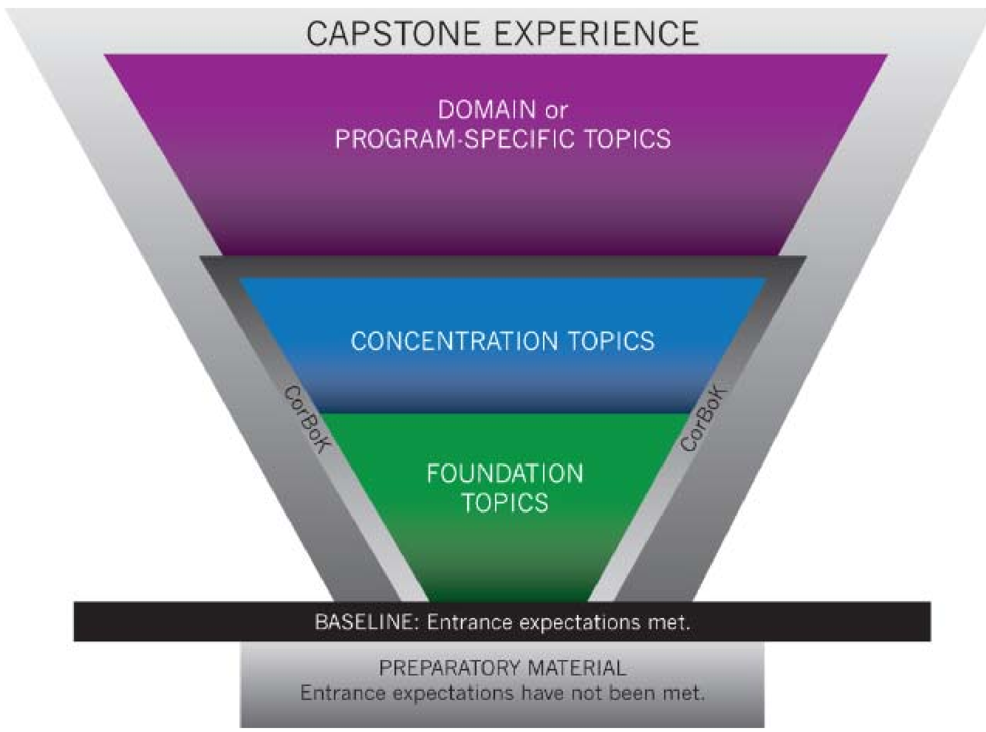

Figure 1. Curriculum Architecture of a GRCSE Master's Program (figure developed for BKCASE).

The foundation and concentration material is intended to constitute about half the coursework for the degree, leaving the remaining portion of the program for program-specific work; however, this does not preclude the CorBoK topics from being addressed in other parts of the curriculum, including the capstone experience. The actual material comprising the foundation and concentration is listed in Chapter 7, where each type of knowledge is listed as a topic from the SEBoK and most but not all topics are assigned to foundation or concentration.

[3] The natural sciences include biology, chemistry, physics, astronomy, and earth sciences.

Summary of Core Body of Knowledge

The foundation and concentration knowledge areas together form the Core Body of Knowledge (CorBoK) for GRCSE. The CorBoK identifies the topic areas that address the skills, knowledge, and abilities a student should learn to achieve the expected outcomes upon receiving a master's degree in systems engineering. The knowledge in the CorBoK includes topic areas covered in the *Guide to the Systems Engineering Body of Knowledge* (SEBoK) v0.5 (Pyster and Olwell (eds.) et al. 2011).

For each knowledge element, a level of mastery is defined using Bloom's taxonomy. These levels are captured in tables in Chapter 7. Each GRCSE-aligned program is free to aggregate the knowledge into individual courses as they see fit.

As the SEBoK evolves, so too will the CorBoK.

Summary of Assessment

Chapter 8 discusses methods for assessing the achievement of program objectives and outcomes, and provides examples. The approach is summarized in Figure 2.

Figure 2. Educational Requirements and Assessment 'Vee' Model (figure developed for BKCASE).

Other Material

The document also discusses how GRCSE will evolve and contains placeholders for material to be further developed in v0.5, including a competency-based curriculum development approach and a discussion of program emphases. The appendices summarize the survey results, Bloom's taxonomy, a mapping of the recommended outcomes to the CorBoK, and detailed material on program assessment and evaluation.

This page intentionally left blank.

Table of Contents

List of Figures

List of Tables

This page intentionally left blank.

1. Introduction

1.1 Systems Engineering Background

Systems and the processes and practices required to develop them are critical to the operation of the modern world. The International Council on Systems Engineering (INCOSE) defines systems engineering (SE) as

> *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, and then proceeding with design synthesis and system validation while considering the complete problem: operations, cost and schedule, performance, training and support, test, manufacturing, and disposal. Systems engineering considers both the business and the technical needs of all customers with the goal of providing a quality product that meets the user needs. (INCOSE 2010b)*

SE principles and practices are essential for the development of large, complex, or trustworthy systems, whether they are products, services, or enterprises. SE enables a contemporary lifestyle reliant on high-performance systems in every sector, including medical devices and services, automobiles, aircraft, power supplies, communication systems, and entertainment systems. Modern design philosophies demand systems thinking in order to embed intelligence in products, interlink them deeply with software and people, and ensure that desired capabilities are reliably provided.

Systems engineers are responsible for ensuring that the right system development tasks are performed effectively and efficiently so that high-quality systems can be delivered and sustained. Systems engineers work with a variety of other professionals to create a system and support it throughout its life cycle. These other professionals contribute skills in various engineering disciplines, systems science, project management, and many other fields, that support the effective conduct of projects within enterprises. The role of the systems engineer includes

- understanding the intended purpose, operational context, and concept of use of the proposed system;
- appreciating the interests, purposes, and values of multiple stakeholders and the ability to combine these into a coherent representation of the project environment;
- understanding the technology which could be applied in the system;
- appreciating the life cycle implications of systems and the ability to incorporate life cycle perspectives into systems design; and
- evaluating, selecting, and developing system solutions to satisfy customer needs and project objectives.

The knowledge, skills, and abilities (KSAs) required to perform this role are diverse and may differ from those needed in other engineering disciplines. As SE continues to mature, curricula that foster these

KSAs is becoming critically important. Such a curriculum should enable students to continue their professional development after graduation and eventually move into engineering and technical leadership of large, complex projects.

1.2 Systems Engineering Education

The roots of SE education reach back to the development of SE as a distinct field during the 1950s. A few years later, *IEEE Transactions on Education* (previously *IRE Transactions on Education*) published a special issue on SE education. Topics included the description of programs (Wymore 1962, 108-112; Showers 1962, 113-115; Reitman 1962, 119-121; Nara 1962, 105-108), the means of emphasizing systems level optimization (Matchett 1962, 85-86; Howard 1962, 78-81; Truxall 1962, 82-85; Linvill 1962, 64-67; Hamming 1962, 76-78), the human interface to equipment (Karlin 1962, 71-75), and the need for students to be introduced to thinking about engineering at a systems level rather than a component or equipment item level (Brainerd 1962, 55-56).

SE education has grown since the 1960s, with many institutions worldwide now offering degree programs in SE at bachelor's, master's, and doctoral levels. These programs comprise two distinct families: those centered on SE specifically and those centered on an engineering domain (such as industrial, biological, or computer) combined with SE. Fabrycky calls these "systems-centric" and "domain-centric" SE programs, respectively (Fabrycky 2010). The two types of programs have different primary purposes, and impose different requirements on the SE aspects of their curriculum, particularly at the bachelor's and master's levels. The *Graduate Reference Curriculum for Systems Engineering* (GRCSE™) project presents a reference curriculum for systems-centric master's programs.

The master's programs most commonly sought by students are professional degree programs focused on developing or improving their skills as SE practitioners, as evidenced by Fabrycky (2010) who reports on 31 master's programs and 14 Ph.D. programs in SE in the US. The popularity of professional master's programs in SE has guided the specific focus of GRCSE on these programs to the exclusion of other programs. Such students are, at the time of their decision to study in a professional master's program, not focused on progressing through to a doctoral program, although some may later decide to pursue doctoral studies. Many students in SE master's programs hold bachelor's degrees in other areas, such as other fields of engineering or science, and approach their master's program in SE as a career development step. In some cases, the students are recent graduates with a bachelor's degree, but little or no real-world SE work experience. Their lack of experience is a challenge in realizing the educational objectives and outcomes identified in GRCSE, and this concern is explored in several places in this document.

1.3 Document Purpose

Despite the importance of graduate education to today's systems engineers, there is no community-accepted recommendation or guidance on what to teach graduate students about SE. GRCSE offers a reference curriculum[4] for a professional master's degree in SE.

Naturally, GRCSE draws on several earlier efforts. In 2007, INCOSE published a reference curriculum framework for a graduate program in SE (Jain and Verma 2007). This framework was the culmination of a two-year effort by a small team to establish a baseline curriculum for a SE graduate program (Squires and Cloutier 2010). In July of that year, the United States Department of Defense's (DoD) Office of the Secretary of Defense (OSD), motivated by challenges in acquiring, operating, and maintaining defense systems, began sponsoring a series of master's-level software and systems curricula. This sponsorship enabled the development of *Graduate Software Engineering 2009* (GSwE2009), Version 1.0 (Pyster 2009), which was released in September 2009, and is now sponsored by the Association for Computing Machinery (ACM) and the Institute for Electrical and Electronics Engineers (IEEE) Computer Society. GSwE2009 provides guidelines and recommendations for a professional master's degree in software engineering (SwE). Recommendations from the INCOSE and the GSwE2009 graduate reference curriculum projects have significantly informed GRCSE.

GRCSE is written to assist developers of new programs, reviewers of existing programs, prospective students, and prospective employers of SE master's program graduates. For each of these stakeholders, GRCSE is a tool to support development, maintenance, updates, or selection of master's programs in SE to meet their particular needs. To that end, GRCSE is intended to

- enable program developers and maintainers to improve existing SE graduate programs from the viewpoint of universities, students, graduates, employers, and systems customers and users;
- assist the development of new master's SE programs by providing guidelines on curriculum content and advice on how to implement those guidelines;
- provide a framework to guide the deliberations of strategic advisory boards, which are often established to assist universities in the appropriate design of their programs;
- support increased enrollment in SE programs by increasing the value of those programs to potential students and employers; and
- assist in an overall understanding of the diversity of available SE educational programs to support prospective students and employers in gauging the suitability of a particular program for their individual purposes.

[4] A *reference curriculum* is a set of outcomes, objectives, entrance expectations, architecture, and a body of knowledge that provides guidance for faculty members who are designing and updating their programs. That guidance is intentionally flexible so that faculty members can adopt and adapt it based on local programmatic needs. A reference curriculum is not intended to be used directly for program accreditation, but certainly can inform faculty members who wish to design a curriculum so their program can eventually be accredited.

1.4 Guidance for Developing GRCSE

The GRCSE team began by first outlining principles and assumptions to guide the work. These were enumerated in the first version of GRCSE, Version 0.25, released for review from December 2010 through February 2011. They are reproduced in Appendix A.

To briefly summarize the guiding assumptions and how they are reflected in GRCSE:

- SE is a distinct discipline and contains a rich body of knowledge. Therefore the *Systems Engineering Body of Knowledge* (SEBoK), Version 0.5, serves as an important input to GRCSE (Pyster and Olwell (eds.) et al. 2010).
- SE interacts with other disciplines, some of which provide important foundation concepts for SE. These disciplines are integrated into GRCSE as appropriate.
- Existing SE programs are diverse. The GRCSE team conducted a survey of SE master's programs that confirmed that there is considerable diversity among existing programs (see Appendix B for summary results of this survey).
- SE is by nature a practical discipline and therefore students must learn how to integrate theory and practice. This is discussed in the outcomes and objectives of GRCSE.

1.4.1 GRCSE Scope

GRCSE is aimed at university education leading to a professional master's degree in SE; that is, a degree intended for someone who will either enter the workforce as a systems engineer, or who is already in the workforce and seeks to gain more formal education in SE to advance his or her career. GRCSE does not target graduate programs for those seeking a doctoral degree and a career in research or education, and does not discuss developing a student's ability to perform research. Rather, GRCSE discusses programs that develop skills to perform SE in professional roles. However, since GRCSE is designed to be tailored and incorporates freedom for university-specific content, university curricula can include preparation for doctoral studies if desired.

GRCSE respects and accommodates the tremendous diversity in the markets that universities serve, the educational systems under which universities operate, the sizes of student bodies and faculty, accreditation programs, and other factors that affect program content and delivery. Therefore, GRCSE is a broad set of recommendations intended to guide universities in building and updating their graduate programs. It is not a standard, but rather a tool for SE curriculum designers.

GRCSE may prove helpful for purposes not explicitly intended, such as the development and maintenance of the SE aspect of degree types not specifically addressed in GRCSE. While valuable, providing such benefits is not the purpose of GRCSE.

1.4.2 Global Applicability

GRCSE is designed for worldwide use. Globally, education systems differ significantly, with each country or region imposing its own regulatory environment and cultural norms. Regulation may constrain the nature of programs that can be offered at the master's level or may constrain course offerings or some other aspect of programs. Culture or regulation may influence whether students proceed directly from undergraduate to master's level studies or expect to gain some career experience between the two education stages. The choice between face-to-face, online, or hybrid delivery, and the administrative decomposition of programs, can also be affected by regulation and culture.

1.4.3 Accounting for Multiple Education Pathways

The permutations of education program responses to regulation and culture are too numerous and diverse for a reference curriculum to meaningfully respond to each. The diversity of possible education pathways is illustrated in Figure 3.

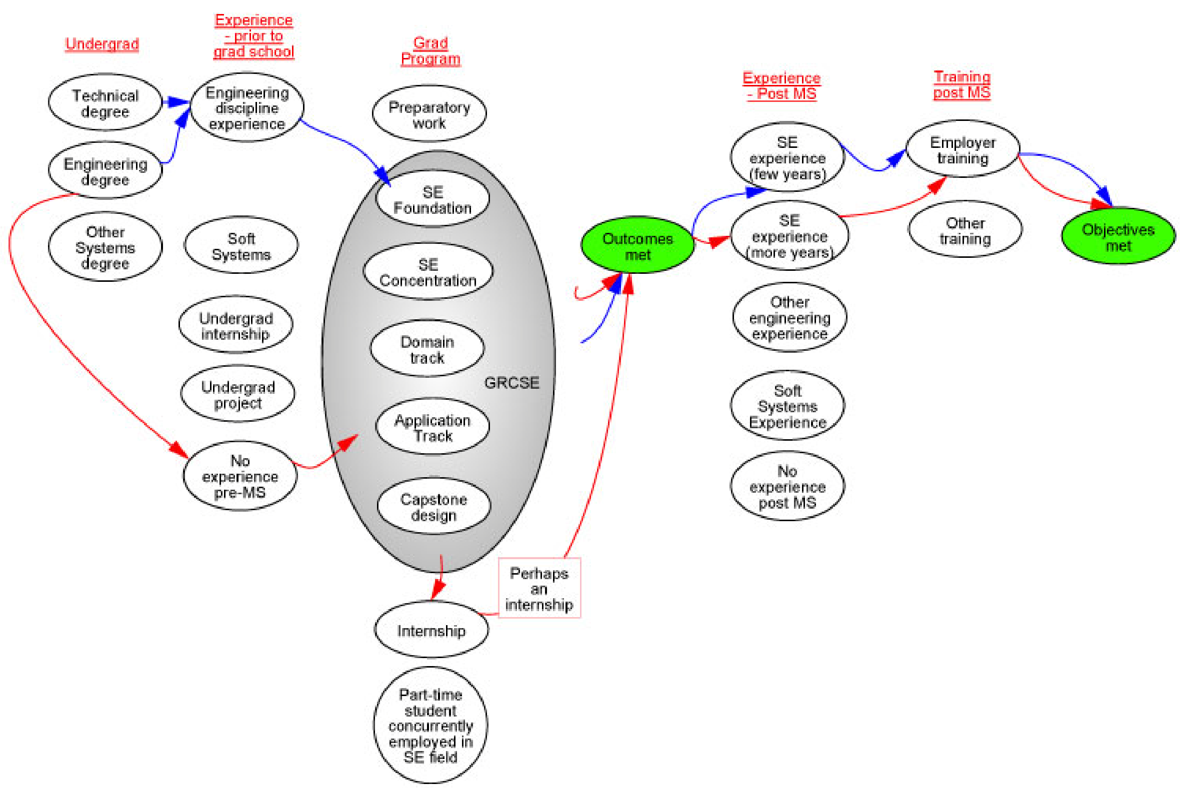

Figure 3. The Diversity of Education Pathways Accommodated in GRCSE (figure developed for BKCASE).

The GRCSE authors considered directly addressing two principal common paths. On the first path, a student proceeds from undergraduate study to a professional master's degree with no significant experience between the two programs (shown in red in Figure 3). On the second path, the student gains significant experience between the bachelor and master's degree level studies (shown in blue in Figure 3). Whether the student obtains work experience between completing an initial university qualification and commencing master's level studies is certainly one of most important factors in professional master's level study. This is true regardless of the underlying causes of pathway choices by individual students.

The GRCSE authors considered the relevance of each of these pathways in different educational settings and discussed the relative popularity of the red and blue paths. In the US, most students follow the blue path: they take jobs after their undergraduate education before pursuing a graduate degree in SE. In Europe, most students follow the red path: immediately after earning their undergraduate degree, they seek a master's degree in their undergraduate field, typically a traditional engineering field like mechanical, civil, or electrical engineering. They then get work experience and return for a second master's degree in SE. Few obtain an undergraduate SE degree or study SE for their first master's degree.

Despite their differences, then, most students in Europe and the US do acquire work experience prior to entering the SE master's program. For this reason, GRCSE 0.5 recommendations address students following the blue path. If GRCSE 0.5 review comments indicate that the red path should also be addressed, Version 1.0 will include relevant recommendations.

1.5 GRCSE Outline

The wider context in which education occurs includes desires and needs of students and potential employers along with university-specific requirements, objectives, and mandates, as shown in Figure 4 (below). These provide input to objectives, which are expectations of what a student will have achieved by a period of time after graduation. Objectives influence outcomes, which are expectations of what a student will be able to do immediately upon graduation from a program. Program architecture defines the topics and experiences that must be covered for students to achieve the outcomes. Outcomes must be assessed, which in turn informs the design of the overarching curriculum. Meanwhile, the available pool of potential students informs the program's expectations about student backgrounds. Student ability to achieve outcomes partly depends on background; understanding this informs the program's approach.

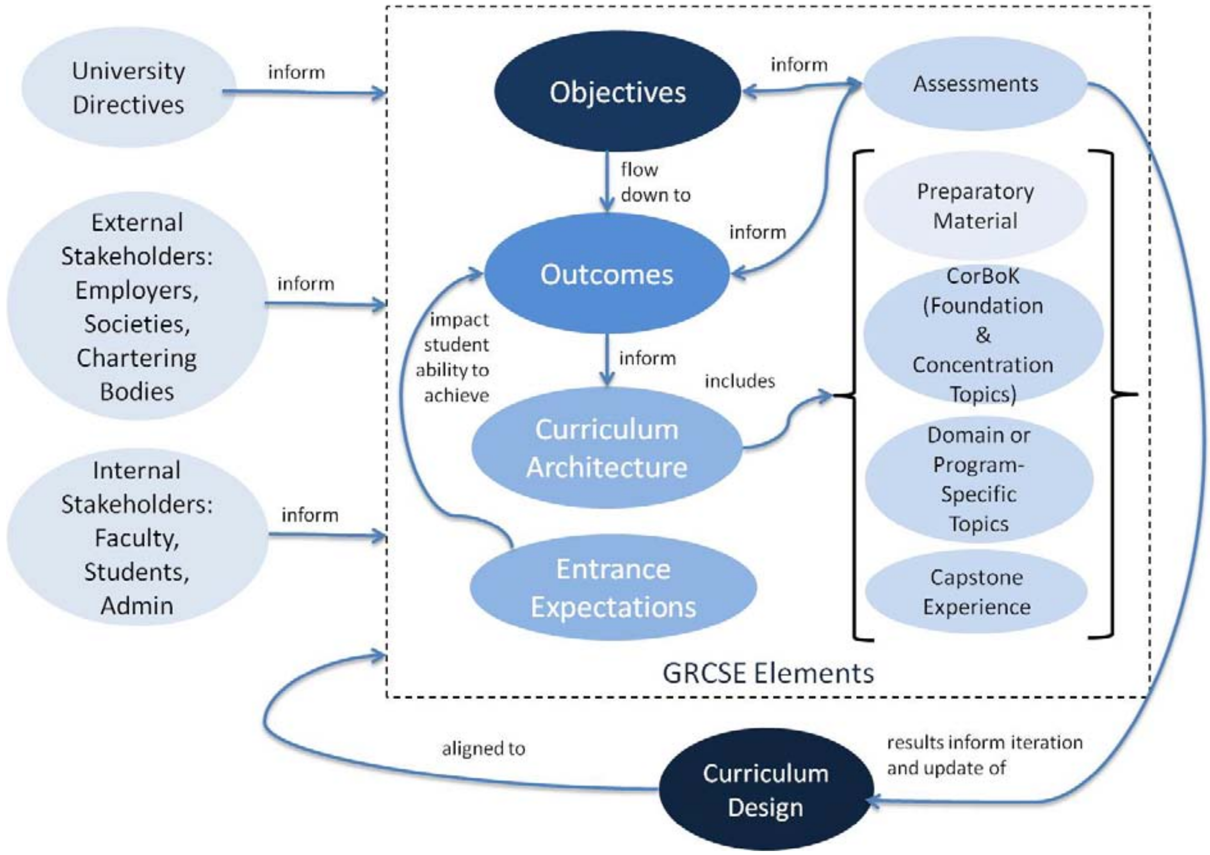

Figure 4. GRCSE Curriculum Elements in the Context of Curriculum Development Influences (figure developed for BKCASE).

All of these interactions are important aspects of curriculum design. GRCSE specifically focuses on the areas contained within the dotted line of Figure 4. This information is presented in nine chapters:

- Chapter 2 describes the objectives students should attain three to five years after graduation.
- Chapter 3 states the outcomes that a student is expected to achieve immediately upon graduation.
- Chapter 4 details the background students are expected to possess before entering a master's program.
- Chapter 5 presents curriculum architecture for structuring a SE program and a common mechanism for communicating the components of an institution's SE graduate level curriculum. The GRCSE architecture shows the relationships and overlap between the *Core Body of Knowledge* (CorBoK), domain or program-specific topics, capstone experiences, and leveling courses.
- Chapter 6 describes the CorBoK, which includes both the foundation (topics which should be learned by all students) and the concentrations (topics which should be covered by students focusing on a specific SE role). The CorBoK is intended to cover no more than 50%

of the total knowledge conveyed in a graduate program. Making the core knowledge 50% of the program instils critical commonality among programs. Employers will have more information regarding what students graduating with a master's program have learned and are capable of doing. Restricting the CorBoK to no more than 50% encourages significant variation among programs while simultaneously building on the common foundation. This ensures an opportunity for the student to develop a deeper knowledge in topics of particular interest, such as requirements elicitation and analysis or system architecture.

- Chapter 7 includes guidance on implementation, focusing on using GRCSE as a tool for curriculum development and revision. It includes considerations for tailoring GRCSE recommendations to fit a program's specific needs in terms of stakeholder requirements and environmental constraints. For example, to align with the GRCSE recommendations, a program should enable its students to achieve all thirteen outcomes listed in Chapter 3, but that program could also add several outcomes that are specific to that program's clientele, faculty interests, and other relevant factors.
- Chapter 8 provides guidance for developing assessment rubrics to ensure that graduate programs achieve their intended outcomes. This chapter is built on the general discussion of the relationship of student learning and assessment in Appendix E.
- Chapter 9 explains the intended evolution and long-term support of GRCSE.

Together, these chapters present an integrated vision of a reference graduate curriculum in SE. Because each university will tailor and adopt GRCSE recommendations around its unique educational philosophy, legacy, faculty strengths, and many other factors, GRCSE does not prescribe a particular set of courses[5]. However, GRCSE 1.0 will examine how several existing SE programs implement GRCSE recommendations.

[5] The term course refers to a collection of materials, exercises, and assessments for which academic credit is awarded. A program is a collection of courses leading to a degree – the specific interest here being a master's degree. Often, programs have one or more specific orientations called tracks or focus areas that allow a student to specialize in an area of interest.

2. Process for Developing Objectives

2.1 Objectives

This chapter describes the process for setting objectives of curricula that align with the *Graduate Reference Curriculum for Systems Engineering* (GRCSE) recommendations. According to the Accreditation Board for Engineering and Technology (ABET[6]),

> *program educational objectives are broad statements that describe what graduates are expected to attain within a few years of graduation. Program educational objectives are based on the needs of the program's constituencies.* (ABET Engineering Accreditation Commission 2010, 2)

An educational institution sets its program objectives after considering the needs of the program stakeholders.

Program outcomes (see Chapter 3) are narrower statements that

> *describe what students are expected to know and be able to do by the time of graduation. These relate to the skills, knowledge, and behaviors that students acquire as they progress through the program.* (2010)

Program outcomes are set by the educational institution to define the exact attributes of the program graduates at the time of their graduation. The relationship between program objectives and outcomes is represented in Figure 5.

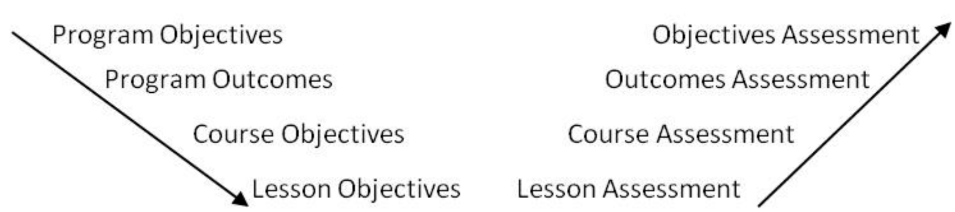

Figure 5. The Relationship Between Program Objectives and Outcomes, and the Corresponding Assessments (figure developed for BKCASE).

GRCSE is intended for global use, and ABET, while international, is only one of several accrediting bodies in the world for engineering programs. Nevertheless, the ABET definitions are useful. The ABET definitions of objectives and outcomes are consistent with the terminology used internationally and are broadly similar to the language used in the discussion of curricula in many fields. Also, as systems engineering (SE) increases its maturity as an engineering discipline, defining its curricular design in the

[6] The reference to ABET is used here to support the definition of objectives. However, as stated in Guiding Principle 18 in Appendix A, while GRCSE *could* inform accreditation, certification, or licensing, it is not intended to be used directly for any of these purposes.

terms used by other engineering disciplines will assist in the acceptance of SE. Consequently, ABET definitions and recommendations are referenced in GRCSE.

Program objectives are informed by multiple sources, such as organizations that hire engineers, competency models provided by these organizations, educator-distributed survey results, etc. Organizations that hire engineers analyze their engineering workforce requirements and document the results, in part, in competency models (see Appendix F) that define what accomplishments are expected from their engineers three to five years after graduation (this applies to graduates of both bachelor's and master's level programs). In turn, these organizations, individually or collectively, may provide their competency models to educational organizations that may then use them wholly or in part to develop program objectives. This may be a formalized process, with customer organizations recognizing or even directly sponsoring programs against agreed objectives. Alternately, a more market-led approach may have educators using surveys or other means to identify program objectives that will attract potential students. Most programs have at least an informal sense of the needs of the potential high-volume employers of their graduates. Few programs have formally specified these objectives, however, and GRCSE explicitly seeks to increase this practice.

The design of curricula is primarily predicated on achieving the program objectives. Consequently, the program outcomes selected should directly support the matriculated engineer in achieving the program objectives. The program outcomes subsequently drive the course selection for the curriculum and the learning objectives for each course (see Figure 5 above).

Program objectives and program outcomes are crucial statements about curricula. Accordingly, the GRCSE team conducted a global survey of graduate SE programs (see Appendix A) to collect these programs' objectives and outcomes. The results of the surveys were used to shape the suggested objectives and outcomes in Chapters 2 and 3.

Objectives are one way to distinguish programs from one another. For example, in a program that may typically place graduates into the aerospace industry, the set of objectives would address the workforce needs of the aerospace industry and the associated outcomes would describe how the program prepared graduates for that objective. Another program may provide system engineers to the communications industry, and its objectives and outcomes would reflect the workforce needs of that industry. Programs that service the government acquirer would have at least some different objectives than those programs that service the government provider.

2.2 Setting Objectives

A large, mature engineering organization often has a formal workforce development model. These models vary by organization and product line, but have common features. For example, formal models are designed to assure that the engineering workforce develops the proper set of competencies to support the organization (see Appendix F).

Competency models specify the required knowledge, skills, and abilities (KSAs) desired of the various classes of employees at different stages in their careers. These KSAs are often listed in position descriptions. Organizations use undergraduate education, graduate education, in-house training, external training, and work assignment policies to provide the means for employees to obtain the required level of proficiency in specified KSAs. Workforce development plans use the competency models to map the desired KSAs against the means used by the employees to obtain them.

Employees are often certified, either internally or externally, or licensed. This certification or licensing serves as a designation that they possess KSAs to perform certain duties in the organization. If a program intends to channel students into a certification or licensure regimen, the appropriate educational requirements should be incorporated into the objectives. Licensure may have other requirements besides education and it is worth mentioning that these other requirements, such as experience or peer review, may be beyond the scope of the educational program. However, it is useful for the designers of an educational program to be cognizant of these other requirements so that the educational program will provide the student with the foundation, to the extent feasible, to achieve the other requirements. In this case, an organization communicates with internal or external educators and trainers to define what current or potential employees of their organization need to learn.

Accreditation agencies have their own specialized language, and those terms (objectives and outcomes) are often used to communicate with educational institutions. Educational institutions establish industrial advisory boards (IABs), or similarly named entities, for their stakeholders to influence the design of their programs. As an example, ABET requires that the needs of the educational institution's constituencies be addressed in its program objectives and the use of these advisory boards is a common method for constituencies to provide input relative to their needs. In most of the United States (US), one of the prerequisites for a professional engineer's license is completion of an ABET-accredited engineering program (The Professional Engineer 2011). In the United Kingdom, registration as a chartered engineer usually requires completion of an accredited degree at either the bachelor's or master's level (Engineering Council 2011). Universities communicate with their stakeholders, usually through these advisory boards, and consider their input carefully when formally defining program objectives. However, the objectives, like the outcomes, are set by the faculty.

2.3 Example Objectives

To illustrate the implementation of objectives for a program, three examples are provided: two from the programs in the US and one from a program in France. As other graduate institutions define objectives, future versions of GRCSE will expand the set of exemplars.

The US program examples come from a survey of graduate programs (Appendix B) that discovered only two instances of SE programs in the US that had formal program objectives. Both US programs were subject to ABET accreditation at the graduate level, which requires the publication of objectives, and both US schools serve the US defense enterprise. Since ABET only accredits an educational institution's lowest degree-granting engineering program, and these schools granted only master's and doctoral degrees, they provide a unique reference point for objectives for a SE master's degree.

The objectives of the first school are listed below:

- Graduates will rise to positions of technical and/or programmatic leadership within capability planning, system acquisition, and/or sustainment organizations. Examples of leadership positions include technical director, division chief, chief engineer, and system program manager or director.
- Graduates will employ SE methods and tools across the Department of Defense (DoD) acquisition life cycle. Examples of SE methods and tools include decision analysis, trade studies, risk management, requirements management, architecture definition or evaluation, and capability assessment.

The objectives of the second school are:

- Leadership – Students will be provided with an educational foundation that prepares them for leadership roles along diverse career paths.
- Program Management – Students will be provided with an educational foundation that prepares them for assignments related to research, design, development, procurement, integration, maintenance, and life cycle management of systems for defense and national security.
- Operational Utilization – Students will be provided with an educational foundation that allows them to understand the capabilities and limitations of military SE and to effectively employ SE in diverse military settings.

Both of these US programs are specialized in the sense that they serve the US defense enterprise; their objectives are not immediately transferable to other programs. However, they do serve to illustrate that objectives are "broad statements that describe the career and professional accomplishments that the program is preparing the graduates to achieve" (ABET Engineering Accreditation Commission 2010, 2). From these objectives flow the specific program outcomes of these schools.

The third and final example of program objectives comes from a university in France. The following example is translated from the original French by one of the GRCSE authors:

- The aim of the master is to propose an initial training for future system architects and project leaders in the field of specification, design, deployment, and maintenance of complex Air Traffic Management systems. At the end of their training, students should be able to take up the main challenges of these systems, such as safety, efficiency, delay, cost, and environmental impact.
- (This university) educates both future non-governmental and governmental engineers.
    - Non-governmental engineers are destined for a career within companies in aeronautic (e.g., EADS, Airbus, and Air France), in electronics (e.g., Thales, Rockwell Collins, etc.), in software engineering (e.g., as IBM, Steria, etc.), in space (e.g., CNES, Astrium, etc.), and in many other domains. When they are working for these companies, engineers will be able to hold the following positions: development engineer (with the objective to participate on the development of new systems), engineering consultant (with the objective to be able to make links between

research advances and practical industrial application), and/or project leader (with the objective to be able to harmonize the technical, financial, and commercial aspects necessary for the success of industrial projects).

- Governmental engineers will be employed by Direction Generale de l'Aviation Civile (DGAC), which is a French Civil Aviation Authority, at Ingénieur des Etudes et de l'Exploitation de l'Avaiation Civile (IEEAC), which stands for Engineer Studies and the Operation of Civil Aviation, in its various departments holding the same positions.

2.4 Generic objectives

GRCSE proposes a generic set of program objectives. The objectives are offered as a starting point that should be tailored by each institution to reflect their vision for graduates informed by each institution's stakeholders. The objectives aim at the time frame of three to five years after graduation, thus paralleling the ABET approach.

Three to five years after graduation, program graduates will

1. Effectively analyze, design, or implement feasible, suitable, effective, supportable, affordable, and integrated system solutions to systems of products, services, enterprises, and system of systems, throughout the entire life cycle or a specified portion of the life cycle. This could be tailored by explicitly stating the types of systems that graduates develop and a given domain (e.g., aerospace).
2. Successfully assume a variety of roles in multi-disciplinary teams of diverse membership, including technical experts and leaders at various levels.
3. Demonstrate professionalism and grow professionally through continued learning and involvement in professional activities. Contribute to the growth of the profession. Contribute to society through ethical and responsible behavior.
4. Communicate (read, write, speak, listen, and illustrate) effectively in oral, written, and newly developing modes and media, especially with stakeholders and colleagues.

These generic objectives were developed to describe what a graduate would likely accomplish in the near to mid-term in their career (see 1 above) and to parallel those found in other engineering disciplines (see 2, 3, and 4 above). Schools should tailor the objectives to their own program stakeholders.

Employers of the program's graduates provide the context for the objectives of a given institution. They also provide feedback, as requested by the institution during the assessment of its program objectives, regarding how well students were prepared to meet the program's objectives. The feedback is used to adjust program objectives, outcomes, and content.

Suitable measures of these objectives could include successful completion of professional registration, certification, or licensure, depending on the national standards; participation in a professional society; appropriate job progression; and supervisor feedback. In particular, becoming an INCOSE Certified SE Professional (CSEP) would be evidence of meeting several of the objectives.

3. Expected Outcomes When a Student Graduates

Chapter 2 describes the process for setting objectives*.* Building on that process, this chapter describes expected outcomes that define what students should be capable of when they graduate from a program that follows the *Graduate Reference Curriculum for Systems Engineering* (GRCSE) recommendations and satisfies the individual program objectives.

3.1 GRCSE Outcomes Approach

The GRCSE expected outcomes express what is required to commence or enhance professional practice and include a mix of technical and professional focus, reflecting the diverse skills that graduates require to become successful system engineers.

This chapter identifies generic statements of outcomes from which a university could define its own set of relevant and measurable outcomes. Some outcomes, in particular the foundation outcome (Outcome 1), the concentration outcome (Outcome 2), and the topic depth outcome (Outcome 3), are supported by identified levels of Bloom's taxonomy (described in Appendix C) and the CorBok (Chapter 6), and thus can be associated with specific, measurable knowledge, skills, and abilities (KSAs), which are very useful for assessment. Associating similar measures with the less tangible and quantifiable outcomes is more difficult. Where appropriate, some advice on how to do this is included. In addition, Appendix D shows the alignment between the outcomes described below and the knowledge areas in the *Core Body of Knowledge* (CorBoK); these are the expected knowledge areas in the *Systems Engineering Body of Knowledge* (SEBoK) that need to be covered in a graduate curriculum.

The GRCSE team modeled the expected outcomes from a reference curriculum for software engineering: *Graduate Software Engineering 2009 (GSwE2009)* (Pyster (eds.) et al. 2009). For software engineering, GSwE2009 is a reference curriculum targeted at a professional master's degree and has been well received, thus it serves as a useful starting point for approaching outcomes. It is important to note that in standard curriculum development practice, the outcomes are derived from the objectives. As an example, the GRCSE team has provided a mapping between the expected outcomes and the sample objectives found in Chapter 2; this mapping can be found in Table 4, which is located after the outcomes.

3.2 GRCSE Expected Outcomes

Note that the order in which the outcomes are listed does **not** reflect a priority*.*

Upon graduation, students who attend a master's program in systems engineering (SE) that follows the GRCSE recommendation are expected to:

1. ***Foundation*** *– Achieve designated Bloom's levels of attainment for each topic contained within the CorBoK foundation.*

   The CorBoK foundation specifies the topics that all students should learn in a professional SE master's program and a minimum Bloom's level for each topic included (please see Chapter 6 for additional detail). It is a basic principle of GRCSE that all successful SE master's students will have reached these levels of achievement.

Mastering the foundation requires learning principles exemplified through practice. A graduating student will have demonstrated the ability to perform at the specified Bloom's level, which ranges from knowledge (the lowest level) to analysis (the fourth level of the taxonomy).

GRCSE does not state how the demonstration of achievement at the defined Bloom's level will be performed. Each individual program must decide how to implement the recommendation through its design of the assessment processes. Appendix D offers some general guidance on the kinds of specific learning outcomes and associated assessment approaches that could be used to assess student achievement. Chapter 8 provides actual examples of outcomes and objectives developed as part of the assessment process.

2. ***Concentration** – Achieve designated Bloom's levels of attainment for each topic contained within one of the CorBoK concentrations, as appropriate for the type of master's program or for an individual student's interest.*

   GRCSE recognizes that there are different viewpoints on the role of systems engineers and these are reflected in different profiles of the core knowledge and skills. For example, the knowledge and skills will differ between those who will use SE knowledge to make acquisition decisions and manage SE teams and those who will perform SE technical activities within a team. While all SE graduates should have mastered the foundation (Outcome 1), they may need different additional skills for their SE specialization. Each CorBoK concentration specifies additional topics and/or Bloom's levels for some topics, which will apply to different types of master's programs.

   A program that follows GRCSE recommendations will need to decide early in the curriculum development or update process which of the concentrations it is going to include. Mastering a concentration requires learning principles demonstrated through academic practice. For guidance on the expected levels of attainment for the CorBoK concentrations, please see Chapter 6.

3. ***Topic Depth** – Achieve a Bloom's synthesis level of attainment for at least one topic from the CorBoK (either foundation or concentration).*

   Synthesis, in Bloom's taxonomy, is the "ability to put parts together to form a new whole" (adapted from Bloom et al. 1956). This involves the use of existing ideas to create new ones, generalizing from facts, relating knowledge from several areas, prediction, and drawing conclusions. It may also involve the adaptation of "general" solution principles to the embodiment of a specific problem. A student needs to delve deeply into at least one topic from the CorBoK (see Chapter 7); such depth strengthens the student's analytic skills and enables the student to solve difficult problems in at least one area.

   The selection of a topic of deeper understanding may be standardized by the program in order to fit with its specific objectives in a particular domain or sector. Alternatively, the topic may be selected by a student as an area of specialization that can be achieved through program electives. It is more likely that areas of depth will be taken from the CorBoK concentrations relevant to a program. However, the ability to demonstrate synthesis level ability in an aspect of SE is an important outcome in its own right, and will remain valid even if the student does not go on to practice in that aspect.

4. ***Application Domain*** – *Demonstrate the ability to perform SE activities in one application domain, such as defense, aerospace, finance, medical, transportation, or telecommunications.*

   This domain outcome involves understanding how differences in a domain manifest themselves in both a system and the engineering of that system. This outcome also includes the ability to learn a new application domain. This incorporates an understanding of specialized terminology, technology, methods, tools, and constraints that are unique to the chosen application domain.

   The domain outcome does not require a student to become an expert in an application domain, an achievement that normally takes many years of experience and education. However, SE only becomes tangible when practiced within the context of a domain, where it brings real value and where systems engineers face, on a daily basis, the characteristics and peculiarities of that domain. Priorities, vocabulary, paradigms, technologies, tools, and a myriad of other factors vary from domain to domain and SE graduate students should be exposed to the specifics from at least one domain.

   As a reference curriculum, GRCSE gives each program the flexibility to emphasize its defining characteristics. Nevertheless, depth in an application domain requires knowing how to apply several of the relevant, significant tools and technologies. Therefore, it is strongly recommended that students be required to demonstrate their capability to apply their SE knowledge to at least one domain through prior work experience, projects, and the capstone experience.

5. ***Specialty*** – The *Application of SE principles to address a specialty such as security, agility, or affordability*, *or such as safety-critical or embedded systems*.

   Application includes understanding how differences in specialties manifest themselves in both engineering the system and in the function of the system itself, and includes the skills required to learn a new application type or specialty.

   It is possible that a specialty may be closely associated with a particular application domain. For example, security and privacy are typically extremely important in financial transactions, yet, for an automobile braking system, safety is likely the most important property in developing a solution. Development standards are very important in defense applications, but less important in systems used to create special effects in movies. However, the specialty does not have to be linked to the mastery of an application domain. A program may choose to couple application domain and specialty or may allow students to select an application domain and specialty of interest, even if the speciality is not emphasized in a given domain. Coupling the two may provide students with the opportunity to delve more deeply; if the two are not coupled, students may have more difficulty allocating appropriate time to each.

6. **Realism** – *Comprehend and appreciate the challenges of applying* SE *to realistic problems throughout the system life cycle.*

   GRCSE advises that as a minimum, students should be able to support relevant SE activities across all aspects of a well-bounded project throughout the complete product life cycle. The graduate will be knowledgeable of the overall systems approach at each life cycle stage and have an understanding of all SE activities relevant to the life cycle.

   It is important that students be given sufficient opportunities to understand how activities at different stages of the life cycle come together to contribute to project success, as well as to experience representative real-world issues, such as how to communicate with stakeholders and the impact of change on both technical and non-technical system issues.

   The presence of one or more capstone experience(s), combined with group projects rather than individual activities, such as working on a thesis, is of considerable importance for a student to develop real experience. It offers students the opportunity to address a realistic problem and demonstrate their ability to bring together topics from a variety of courses and apply them effectively. This should also include the ability to offer reflections on their achievements. Universities will wish to tailor this outcome to the stated objectives of their program and the associated professional roles they expect their students to take.

7. **Teamwork** – *Perform as an effective member of a multi-disciplinary team, effectively communicate both orally and in writing, and lead in one area of system development, such as project management, requirements analysis, architecture, construction, or quality assurance.*

   Students need to complete tasks that involve work as an individual, but also must complete many other tasks that entail working with a group of individuals. For group work, students should be informed of the nature of groups and of group activities, as well as their expected roles as explicitly as possible. This must include an emphasis on the importance of such matters as a disciplined approach, the need to adhere to deadlines, how to communicate both orally and in writing, and how teams are evaluated as a whole. Students should have an appreciation of team dynamics and leadership techniques, and be able to lead at least one aspect of system development. It is expected that upon graduation, a master's program student will be able to take a significant role in one or more specific areas of SE. This role can be either technical or managerial, depending upon the focus of the master's program. Experiencing different team roles will help students to understand how they may move forward professionally. Students will also benefit from experiencing the dynamic relationships between team members and from taking appropriate leadership roles to prepare them for potential professional posts.

8. **Problem/Solution Evaluation** – *Be able to evaluate alternative system solution strategies, including how well different solutions relate to the identified problem, and express relevant criteria to ensure solutions are selected against a holistic systems perspective.*

   System solution approaches and relevant solution technologies are continuously evolving in a dynamic discipline like SE. A systems engineer must be able to perform tradeoff studies to identify appropriate solution options, understand limitations and appropriate uses of solution options, and help set solution assessment criteria which cover potential holistic system concerns. Students should be able to act as change agents within their respective professional organizations.

System solution options may be driven by the selection of key technologies; e.g., innovative power supplies, lightweight materials, advanced computing hardware, or software approaches such as data fusion or artificial intelligence. Innovations may also come from the reuse of existing technologies in a new application, or from new manufacturing, testing, support, or marketing approaches. System solutions can also be characterized by the organization and relationships between system components; e.g., centralized vs. distributed control, or from the way a new system is integrated into and interacts with existing systems.

A SE graduate should know how to decide the relative technical and non-technical merits of solution options based on assembled or discovered evidence and effectively advocate for appropriate system choices. Note, however, that in a university setting, a student will likely only be able to demonstrate their potential to be an effective advocate.

This outcome has a strong relationship to the outcome requiring domain experience. The strengths and weaknesses of a system option are generally not absolute, but vary with the application domain and other contextual factors.

9. ***Requirement Reconciliation*** – *Be able to reconcile conflicting requirements, finding acceptable compromises within limitations of cost, time, knowledge, risk, existing systems, and organizations.*

   New requirements routinely emerge during the course of most large or complex projects. Students should engage in realistic exercises that expose them to conflicting and changing requirements. The graduate of a master's program should be able to identify the impacts of such emergence on technical planning, systems architecture, and technical performance, among other considerations. For example, students must be able to identify and elicit the real needs of the customer rather than simply implement a customer's proposed solution that may not fit the real need.

   Once new requirements are identified, a range of appropriate techniques for presenting alternatives and making trades should be introduced as a way of resolving conflicts. The resulting solution should be tailored to meet these conflicting requirements while also following a project's configuration management policies and procedures. Students need to understand the basic techniques available for resolving conflict within the broader SE context and have experience applying these techniques.

   Please note that there is a relationship between this outcome and the outcome on professional ethics (outcome 13 below) as the inability to reconcile conflicts within the policy and legal guidelines can lead to ethical dilemmas.

10. ***Professional Development*** – *Be able to learn new models, techniques, and technologies as they emerge, and appreciate the necessity of such continuing professional development.*

To be effective systems engineers, graduates must be able to grow personally and professionally. In a field as dynamic as SE, lifelong learning is essential to continued success. It is therefore imperative for the graduate student to develop the necessary skills to keep abreast of the latest developments and be able to analyze the advantages and disadvantages of leading edge developments as they relate to the system under design. This includes the ability to evaluate and adapt systems development processes, practices, metrics, and tools to incorporate new advances, as well as the ability to create or assemble satisfactory evidence for these actions. A master‘s program cannot instill the desire for lifelong learning, but can teach the skills that enable lifelong learning.

11. ***Related Disciplines*** *– Comprehend the relationships between* SE *and other disciplines, such as project management, human factors, and other engineering fields as discussed in the SEBoK, and be able to articulate the value proposition of these disciplines for* SE*.*

    SE incorporates skill sets from many disciplines, including more traditional engineering disciplines (electrical, mechanical, civil, etc.) as well as more management-focused disciplines (project management, program management, etc.). It is important that systems engineers possess basic knowledge related to these disciplines and also understand how the SE discipline is related to other disciplines. A student should be able to articulate how SE could and should interact with these disciplines and what common pitfalls may occur when these relationships are not properly managed. This is discussed in "Part 6: Related Disciplines" of the SEBoK (Pyster and Olwell (eds) et al. 2011).

    Given the increasing complexity of modern systems, it is imperative for graduates of SE programs to understand and appreciate the fundamental concepts of project management, their relationship with SE, and the ways in which complexity can be managed in projects. In addition to the general relationships, some disciplines, such as human factors, are now heavily entwined with systems. Human factors deals with complex systems that are generally operated and/or used by humans; in these systems, issues associated with usability and ergonomics play a major role in the success or failure of these systems.

12. ***Software in Systems*** *– Demonstrate an understanding and appreciation of the level of software engineering necessary to develop current and future product*s*, service*s*, and enterprise systems.*

    An adequate understanding of software engineering (SwE) will fundamentally change the way a systems engineer conceives, architects, and implements a system. For example, in today's premium automobiles, software development accounts for 13-15% of the total development cost; safe and efficient operation of hybrid automobiles requires many millions of lines of software working cooperatively in many subsystems (Charette 2009). Failure to adequately understand software behavior is a leading cause of system failure (Charette 2005). Therefore, an understanding of SwE and the unique requirements, considerations, methods, practices, processes, and tools required for good SwE may fundamentally change the way SE is performed. For example, SwE understanding may open additional options for allocating functionality throughout a system.

13. ***Ethics** – Demonstrate knowledge of professional ethics and of the application of professional ethics in decision-making and* SE *practice.*

    Professionals routinely face ethical, legal, and social dilemmas; for example, when is it ethically, legally, and socially acceptable to compromise quality in order to meet schedule? What types of activities constitute a professional conflict of interest or a breach of ethics, law, or social norms? Potential violations of the law are clear, but in most situations, there are no black and white rules for resolving such questions. Professional societies publish ethical codes of practice and graduates should know and apply the relevant codes.

    A SE graduate should have the maturity, knowledge, and judgment to make common professional decisions and take appropriate actions to respond to situations that have ethical, legal, and social implications. Note, however, that in a university setting, a student will typically only be able to demonstrate their potential to make appropriate ethical decisions.

3.3 Alignment between Outcomes and Objectives

It is expected that students of a SE program would be able to demonstrate the above capabilities upon graduation. Chapter 2 presented a set of example objectives for a program, which students should be able to demonstrate three to five years after graduation. The achievement of objectives is dependent upon achievement of the outcomes, life-long learning, and three to five years of on-the-job experience and training. Therefore, it is necessary for the outcomes to support the achievement of the objectives. Table 2 shows the relationship between the recommended outcomes and sample objectives provided in Chapter 2; this is an example of how such an alignment exercise could be conducted during curriculum development. The alignment between outcomes and objectives is classified under the following three categories:

- Strong – There is clear correlation between the achievement of an outcome and the achievement of an objective; if the outcome is not reached by graduation then there is a high probability that the corresponding objective may not be achieved within three to five years.
- Weak – There is marginal correlation between the outcome and the specific objective.

It should be noted that for Table 2 a blank cell indicates no expected correlation between the outcome and the objective.

Table 2. Outcome to Objective Relationship (table developed for BKCASE).

	Objectives			
Outcomes	**1. SE Life Cycle**	**2. Multi-Disciplinary**	**3. Professionalism**	**4. Communication**
1. Foundation	Strong	Weak	Weak	Weak
2. Extension	Strong	Weak	Weak	Weak
3. Synthesis	Strong	Weak	Weak	
4. Application Domain	Strong	Weak	Weak	Strong

	Objectives			
Outcomes	1. SE Life Cycle	2. Multi-Disciplinary	3. Professionalism	4. Communication
5. Specialty	Strong	Weak	Weak	Strong
6. Realism	Strong	Strong	Weak	Weak
7. Teamwork	Weak	Strong	Weak	Strong
8. Problem/Solution Evaluation	Strong	Weak		Strong
9. Requirement Reconciliation	Strong	Weak		Strong
10. Professional Development	Weak	Weak	Weak	Weak
11. Related Disciplines	Weak	Strong		Weak
12. Software in Systems	Weak	Strong		
13. Ethics			Strong	Weak

Chapter 2 provides a process for developing objectives and provides sample objectives, but not a set of recommended objectives because objectives are so program-specific. However, it is still useful for practitioners to understand the alignment between outcomes and objectives. Table 2 is a relatively simple assessment of this correlation, and provides useful insight into the traceability between outcomes and objectives. Chapter 7 provides additional guidance on implementation.

4. Expected Background for Students Entering a Master's Program

For any graduate program, the prior accomplishments of entering students are critical to success. This chapter elaborates on the knowledge, experience, and undergraduate degrees that the *Graduate Reference Curriculum for Systems Engineering* (GRCSE) deems necessary for students entering master's programs in systems engineering (SE).

Attainment of program objectives and outcomes depends, in part, on student capabilities when they enter the program. For example, many topics in the *Core Body of Knowledge* (CorBoK) require a grasp of probability and statistics, and several GRCSE outcomes revolve around mastering the CorBok. A student entering a master's program with little or no understanding of probability and statistics is at a distinct disadvantage in many classes such as risk management, safety engineering, and system verification and validation. Of course, a program may opt to establish entrance requirements that are inconsistent with GRCSE recommendations. Unless the program compensates for the weaknesses of students entering the program, for example by offering leveling courses, such inconsistencies will make it harder for students to achieve GRCSE's recommended objectives and outcomes.

4.1 Findings from the Survey of Existing Programs

A survey of 33 SE master's programs is summarized and analyzed in Appendix B. The survey was international in scope, with data reported from ten counties: Australia, Brazil, Canada, Cyprus, Finland, France, Japan, Singapore, Taiwan, and the United States (US). Although there was great variation in the admission requirements of the programs, the following general commonalities were observed:

- Almost all programs require an undergraduate degree in engineering, natural science, computing, or mathematics.
- Most require a grade point average of at least 3.0 (the equivalent of a "B" average) in undergraduate performance.
- Few programs require special exam performance as a prerequisite for program entry (e.g., the Graduate Record Exam (GRE) in the US or the POSCOMP exam used in Brazil). Some apply their own exam results for student selection.
- About one-third of the programs require between two and five years of experience. Others require significant project work as part of the course of study.
- Most programs require a language proficiency exam for non-native speakers (e.g., in the US, the Test of English as a Foreign Language (TOEFL) and International English Language Testing System (IELTS) are common; in Brazil, proficiency certificates in English and Portuguese are typically required).

These types of entry criteria are not unusual among graduate-level programs in general.

4.2 GRCSE Entrance Expectations

To have a strong probability of achieving all recommended GRCSE outcomes at graduation and eventually achieving program objectives, GRCSE recommends that an entering student satisfy each of three overarching entrance expectations: degree, experience, and language. These expectations are detailed in the remainder of this section. Outcomes are described in Chapter 3, and objectives are described in Chapter 2.

1. *The equivalent of an undergraduate degree in engineering, the natural sciences*[7]*, mathematics, or computer science.*

   GRCSE Preparatory Knowledge (Table 3, below) defines the expected knowledge that would be obtained through the undergraduate or other educational preparation and provides the baseline knowledge expected of students entering a master's program for SE. It is adapted from Table 1 in Chapter 6.2 of GSwE2009 (Pyster (ed.) 2009). The Bloom's levels specified are explained in Appendix C.

   Many existing SE master's programs expect students to have a bachelor's degree in an engineering or scientific field. Such students bring key domain knowledge, significant mathematical skills, and the ability to think analytically, all of which are essential to SE.

   In order to study graduate SE topics, students must possess basic knowledge and skills in mathematics, science, engineering fundamentals, computing, and general education. This is essential for students in both technically oriented and management oriented master's programs in order to learn technical topics effectively. Those managing SE projects (either as developers or acquirers) must be able to interact effectively with a broad spectrum of systems engineers and be capable of understanding, supporting, and evaluating SE activities.

   Programs accepting students lacking the background in Table 3 should either provide opportunities for them to obtain that background (e.g., offering leveling courses) or consider reducing the scope and degree of achievement of the program outcomes (as described in Chapter 3). Also, students with extensive experience in an engineering field may have acquired knowledge and capability in topics listed in Table 3.

   A student may also earn a bachelor's (undergraduate) degree in SE. There are several possible approaches for students with an undergraduate degree in SE who wish to pursue a graduate degree. Among them are the following:

   - a program may waive the basic or introductory SE courses, allowing students to explore more advanced topics or delve more deeply into an area of specialization
   - a student with a systems-centric undergraduate degree could enter a domain-centric program, with greater emphasis on domain-specific learning than general SE learning (e.g., space SE, missile SE, etc.)
   - a student may pursue a master's degree in a related engineering field (e.g., electrical engineering, civil engineering, mechanical engineering, software engineering, etc.)

[7] The natural sciences include biology, chemistry, physics, astronomy, and earth sciences.

Table 3. GRCSE Preparatory Knowledge (table developed for BKCASE). Source: (Pyster (ed.) 2009)

Knowledge Areas	Bloom Level*
Mathematics Fundamentals	
1. Probability and Statistics Basic probability theory, random variables and probability distributions, estimation theory, hypothesis testing, regression analysis, analysis of variance.	AP
2. Calculus and Analytical Geometry Theory and application of differential and integral calculus methods and operations. Study of techniques for describing, representing, and analyzing geometric objects (coordinate systems, algebraic models, graphing).	AP
Science and Engineering Fundamentals	
1. Natural Science Foundations Basic concepts and principles of one of the natural science disciplines (e.g., physics, biology, chemistry, etc.). This will provide students with a technical background in at least one scientific domain area. This preparation should include laboratory work that involves experimental techniques, the application of the scientific method, and comprehension of appropriate methods for data quality assurance and analysis.	K
2. Engineering Fundamentals The nature of engineering, branches of engineering, the design process, analysis and modeling, the role of empirical and statistical techniques, problem solving strategies, and the value of standards. Students should have some level of practical experience, whether through capstones, internships, or course projects. Practical experience should include the application of engineering fundamentals in a specific domain context.	AP
3. Computing Fundamentals Overview of computer organization (computer architecture, operating systems, and programming languages), algorithms, and data structures; software engineering fundamentals (life cycle models, quality, cost, and schedule issues); and development of a software unit (design, coding, and testing).	AP
General Education	
1. Oral and Written Communication Study and application of communication techniques that support development and enhancement of oral and written communication skills.	AP
2. Ethics and Professional Conduct Principles related to the behavior and decision-making of professionals; obligations of professionals to clients and to society; codes of ethics; and social, legal, and historical issues.	CO
* The Bloom's Level ratings are discussed in Appendix C with the following abbreviations used: K – Knowledge, AP – Application, CO – Comprehension	

2. *At least two years of practical experience in some aspect of SE. This experience should include participation in teams and involvement in the life cycle of a system, subsystem, or system component.*

   SE is an engineering field involving the development, delivery, maintenance, and evolution of products, services, and enterprises. SE involves the application of concepts, principles, and techniques to the solutions of engineering problems; such application can be best learned when coupled with actual experience that supports the understanding and appreciation of how these problems are solved. Effective engineering requires learning by doing.

   The richness of the discussions in a graduate class and the sophistication of the analysis that students can perform are driven, in part, by the experience of those students. While there are no known rigorous studies that prove this, the GRCSE authors believe that students with at least two years of practical experience in system life cycle activities have a significantly deeper appreciation of the issues that are addressed in the master's program.

3. *Demonstrated ability to effectively communicate technical information, both orally and in writing, in a program's language of instruction.*

   Language is the primary tool for delivering education. By its nature, SE demands that students have a sound command of the language of instruction. Because SE is a discipline that touches and interacts with so many other disciplines, it is critical to the success of a systems engineer that he or she be able to communicate technical information effectively.

   It is appropriate for universities to ensure that students entering the program have suitable technical language competence to successfully complete the program. This entails using a process to verify that all entering students can write competently and speak effectively in the language of instruction. Verification methods include students submitting writing samples, being interviewed by faculty as part of the admissions process, or taking an exam, such as the GRE in the US. This applies equally to native and non-native speakers.

It is appropriate for universities to consider requiring a certain standard of academic achievement in prior study, or a demonstration of suitability by some other method, such as through an entrance examination or portfolio of work. The university may be motivated by regulatory requirements or some aspect of its own policies. The method of demonstration of ability and the particular standard selected for the threshold of 'acceptable' should be carefully considered to ensure that each is appropriate for selecting students with a high probability of success in the program, without being unnecessarily exclusionary.

The expectations recommended here are *not* admission requirements, which are set by Individual universities and programs. However, deviations from these expectations may require lengthening the program to compensate for gaps in student ability and to enable students to achieve the recommended outcomes found in Chapter 3. A student may compensate for the lack of a formal education by more extensive prior experience, or by taking an internship or pursue some other means to gain that experience while in the degree program. A university can offer leveling courses to provide expected knowledge or skill that some students lack. These adaptations, which are common practices, increase the number of courses that a student must take to earn a master's degree. The curriculum architecture, described in Chapter 5, Figure 6 provides a structure by which a university could address students who do not meet the entry expectations.

This page intentionally left blank.

5. Curriculum Architecture

This chapter defines the reference curriculum architecture. The reference architecture provides a basis to discuss and compare the architectures of actual systems engineering (SE) programs at different universities.

5.1 Influences on the GRCSE Curriculum Architecture

The curriculum architecture is predicated by the *Graduate Reference Curriculum for Systems Engineering* (GRCSE) recommended entrance expectations, the GRCSE recommended student outcomes and objectives, and the following considerations:

- The architecture describes systems-centric SE programs.
- Graduate programs build on knowledge from related undergraduate programs.
- Programs accept students with diverse levels of expertise and competency.
- The *Systems Engineering Body of Knowledge* (SEBoK) is the reference overview of knowledge to be taught to graduate students.
- Programs allow sufficient flexibility for students to specialize in their individual areas of interest.
- Programs support different specializations based on faculty strengths, local markets, and numerous other factors.
- Students require a capstone experience to integrate program outcomes.

5.2 Elements of the GRCSE Curriculum Architecture

The curriculum architecture organizes the topic areas that address the knowledge, skills, and abilities (KSAs) a student should learn in order to achieve the expected outcomes upon graduation with a master's in SE. The architecture does not organize the courses within which the topics will be included in a particular graduate program.

The curriculum architecture is organized into six elements: preparatory knowledge, foundation knowledge, concentration knowledge, domain-specific knowledge, program-specific knowledge, and a mandatory capstone experience.

Each element is described as follows:

- *Preparatory Knowledge* – A student who enters the program without the expected knowledge and experience that is described in Chapter 4 would need to learn the preparatory knowledge at the beginning of their graduate education.
- *Foundation Knowledge* – The foundation is a set of KSAs that applies to all graduates of a SE program; it is intended to provide the common knowledge that all SE master's degree graduates should possess, regardless of educational institution, location, or anticipated future role. The foundation includes those topic areas that should be mastered, to the

specified Bloom's level, by every successful SE graduate. The foundation is part of the *Core Body of Knowledge* (CorBoK) for GRCSE.

- *Concentration Knowledge* – Each student selects a SE related concentration area and learns knowledge in that area. The concentration area specifies additional topics that should be mastered by the student to the specified Bloom's level, or foundational knowledge topics that should be mastered by the student to a higher specified Bloom's level. The concentration areas are dependent on the intended future role of the SE graduate, and support a flexible **curriculum** while also meeting the needs of a comprehensive program. Concentration areas addressed in this version of GRCSE are Systems Engineering Management (SEM) and Systems Design and Development (SDD). The concentration selected is part of the CorBoK for the GRCSE.
- *Domain-Specific Knowledge* – Each program offers one or more domains, such as finance or telecommunications, in which their students can specialize. Each student picks among the choices offered in the program and learns domain-specific knowledge.
- *Program-Specific Knowledge* – Each program selects topics of special interest to it; these are topics that are based on program or particular educational institution focus and/or expertise.
- *Capstone Experience* – Each program expects students to demonstrate their accumulated KSAs in a mandatory capstone experience. The capstone can be implemented through a variety of methods, including individual or team capstone projects or a practicum. The technical work for a project may be distributed through multiple courses, such as by performing system architecture and design at increasing levels of detail. A master's thesis, which meets the expectations for the capstone experience is also a possible implementation.

The foundation knowledge and the selected SE related concentration knowledge together form the CorBoK for GRCSE; the CorBoK is described in Chapter 6. At least 50% of the academic credits for a SE program are expected to be derived from the material in the CorBoK; the academic credit for the capstone experience is additional. However, this does not preclude the CorBoK topics from being addressed in other parts of the curriculum, including as part of the capstone experience.

Figure 6 provides a visualization of the architecture.

Time in Figure 7 starts at the bottom and moves upward but is notional once one reaches the solid black baseline (after completing the necessary preparatory work). That is, a student normally would start out with foundation topics and end with a capstone experience, though this is not always the case. If a student does not already meet the entrance expectations, then he or she is normally required to complete the preparatory material before starting in the actual program.

The area of each graphical shape in Figure 6 is not strictly proportional to the amount of content for each GRCSE curriculum architecture elements; however, the goal is to give a sense of proportion of topic areas. The light gray border around the foundation and concentration topics depicts the entire content of the CorBoK. The capstone experience is intended to show the use of the knowledge gained from each of the topic areas. The intent of the capstone experience is primarily to implement and demonstrate what a student has learned throughout the program. However, it is obvious and expected that a student will have to learn some new things during the capstone experience. The point is that the capstone experience is really on a different dimension than the encompassed topics.

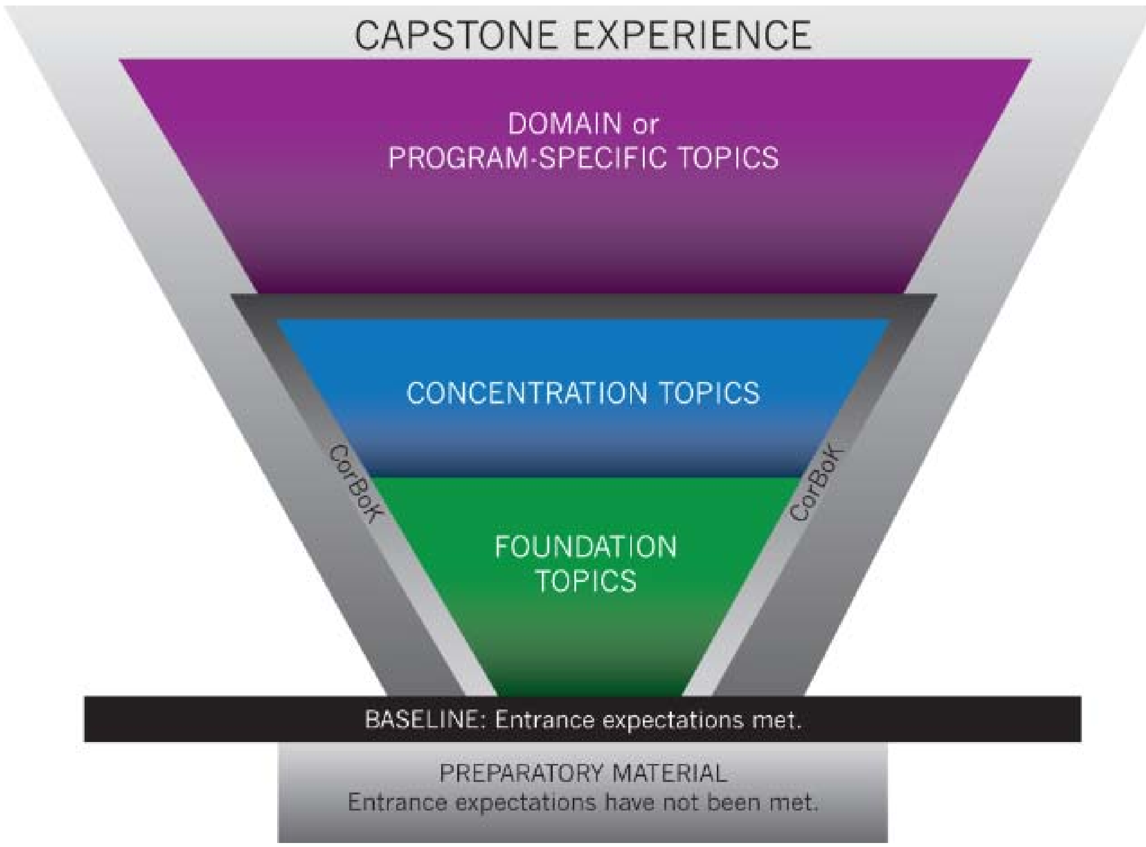

Figure 6. Architectural Structure of Topics in a GRCSE Master's Program (figure developed for BKCASE).

Building on Figure 6, Figure 7 gives an example showing how courses relate to the architecture. The courses shown in Figure 7 are defined as follows:

- Course 1, represented by three circles going along a diagonal on the right side of the figure, covers foundation, concentration, and domain-specific or program-specific knowledge.
- Course 2 integrates foundation and concentration knowledge.
- Course 3 integrates concentration and domain-specific or program-specific knowledge.
- Course 4 focuses on knowledge outside of the CorBoK specific to the domain or program. In most cases this course builds on the foundation and concentration knowledge, but not always.

The courses are depicted as covering topics in more than one knowledge area to show that a program can arrange the topics however it chooses. There is no restriction. A course does not have to be all in one topic area but can contain topics from other areas as well. Also, there is no intent to imply that all courses teaching knowledge in any particular area must be completed before coursework in the next area can begin, although this may be a natural path to follow. A student could, for example, learn some program-specific knowledge before learning all (or even any) of the foundation or concentration knowledge. It is anticipated that the sequencing of courses will be controlled primarily by the prerequisite specifications of each course in a specific program.

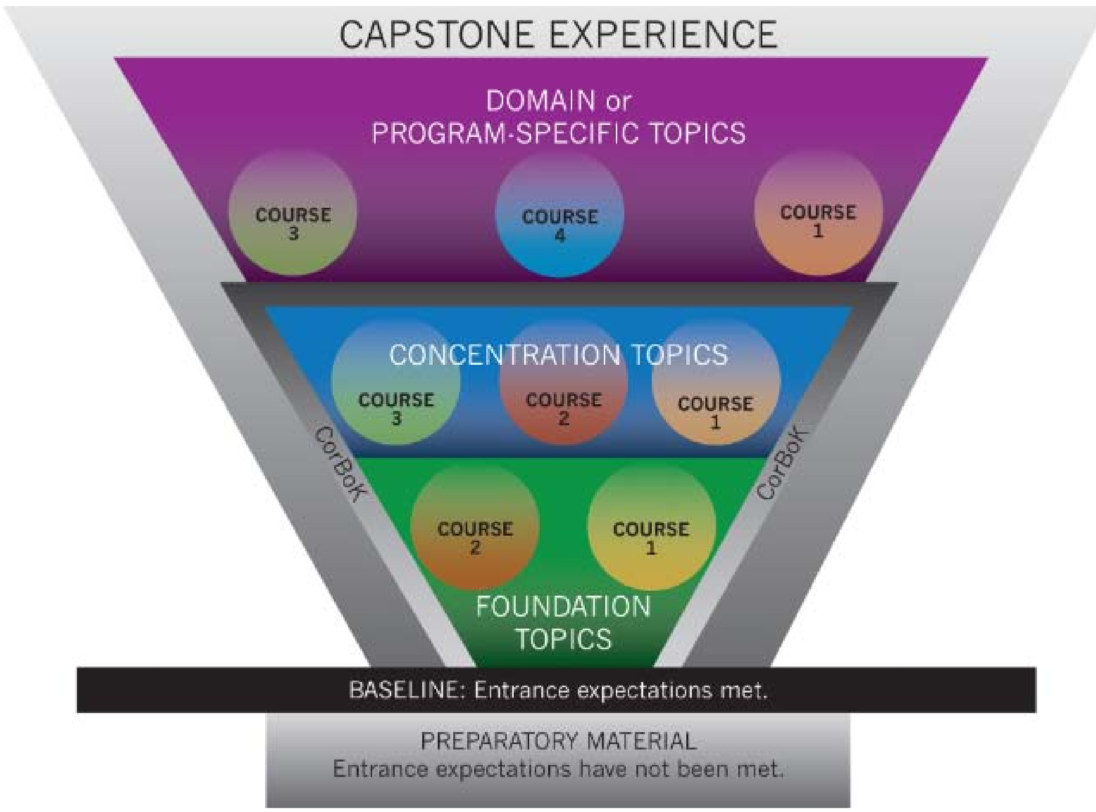

Figure 7. Course Alignment Which May Not Directly Correspond to Topics from Only One Area (figure developed for BKCASE).

6. The Core Body of Knowledge (CorBoK)

In a program that aligns with GRCSE, what are students expected to learn? GRCSE provides its answer in the form of a *Core Body of Knowledge* (*CorBoK*).

Systems Engineering (SE) demands, at minimum

- a basic understanding of all the disciplines required to define, design, implement, integrate, deliver, and manage a system and its components throughout the system life cycle;
- in-depth knowledge of the interfaces between components;
- knowledge required to comprehend the system as a system; and
- the ability to communicate effectively with everyone concerned, including stakeholders.

Given this broad scope, it follows that SE is fundamentally integrative. This quality distinguishes SE from other engineering disciplines. For example, the engineering of a system *component* requires detailed knowledge of the domain contained by the component's interfaces. Mastering this traditional kind of engineering may be best served by learning the intricacies of a specific technology from the bottom up. In contrast, SE deals with the *interfaces between* components, and between the system and its environment. This requires grasping the big picture and integrating interdisciplinary knowledge from a top-down perspective, even while acquiring in-depth knowledge.

Both a broad scope and a need for in-depth knowledge, then, define the nature of SE. The CorBoK is structured in two parts, foundation and concentration*,* to reflect this fact. Foundation knowledge comprises the broad set of topics deemed essential for all systems engineers. Each student is also expected to choose and acquire in-depth knowledge in an area of concentration. See Chapter 5 for the definition of foundation, concentration, and other types of knowledge covered in the curriculum architecture.

The CorBoK identifies the foundation and concentration topics, along with a level of mastery for each, that together address the knowledge, skills, and abilities (KSAs) a student should possess to graduate with a master's degree in SE. GRCSE 0.5 includes two concentrations: systems engineering management (SEM) and systems design and development (SDD). GRCSE 1.0 and subsequent releases may include additional concentrations.

The set of KSAs covered by the foundation provides commonality across all SE master's degree graduates, regardless of school, location, or anticipated future role. Each concentration builds on the foundation by revisiting foundation topics that should be learned in more depth, as well as by adding topics. Concentrations support a flexible curriculum while meeting the needs of a comprehensive program.

6.1 CorBok Development

The CorBoK is expected to comprise approximately 50% of the curriculum (see Chapter 5). Based on the survey of graduate programs conducted at the beginning of the BKCASE project (see Appendix B), the average SE program requires the equivalent of approximately eleven 3-unit semester courses, which expands to approximately 480 contact hours between students and faculty in whatever format the courses are delivered. Hence, CorBoK instruction is presumed to take around 240 contact hours, including examinations and administrative time spent with students. Clearly, this will vary from program to program, but the general guidance of 240 contact hours per student during their education provides the basis for the CorBoK definition.

This section explains both the content of the CorBoK and the process used to develop it. Please note that topics related to domain and program-specific knowledge (see Chapter 5: Curriculum Architecture) are specific to the university and student, and are not included in the CorBoK.

6.2 CorBoK Topics

The SEBoK (see sebokwiki.org) states in its introduction that

> *the Guide to the SE Body of Knowledge (SEBoK) defines and organizes the knowledge of the Systems Engineering (SE) discipline, including its vocabulary, concepts, methods, processes, practices, and tools. It does not attempt to reprint all Systems Engineering knowledge, which is far too large and dynamic to be captured in any single place. Rather, the SEBoK is a guide for the user in finding and understanding the literature about SE that has been separately published in books, articles, websites, and other generally accessible resources.* (Pyster and Olwell (eds.) et al. 2011)

The SEBoK, as a guide to the literature that comprises the body of knowledge of SE, has seven parts, each covering a specific set of knowledge areas, each of which in turn contain a set of topics. There are more than 100 topics overall in the SEBoK. The seven parts are:

- **Part 1 – SEBoK 0.5 Introduction.** This part covers the scope, structure, uses, and evolution of the SEBoK.
- **Part 2 – Systems.** This part describes the characteristics of systems.
- **Part 3 – SE and Management.** This part addresses how SE is conducted and covers life cycle models and processes, management processes, and standards.
- **Part 4 – Applications of SE.** This part covers the application of SE to the development and deployment of products, services, enterprises, and systems of systems.
- **Part 5 – Enabling SE.** This part discusses the enabling of SE at the individual, team, and business/enterprise levels and includes a discussion of ethics, team dynamics, and culture.
- **Part 6 – Related Disciplines.** This part focuses on the relationship of SE to other disciplines.
- **Part 7 – SE Implementation Examples.** This part includes overviews of case studies and vignettes that provide real-world examples of SE activities and provide links back to the concepts covered in the first six parts of the SEBoK.

The CorBoK topics are those in Parts 2 through 6 of SEBoK and are listed in Table 4 through Table 8, which define the CorBoK. As the SEBoK evolves, so too will the GRCSE topic list.

6.3 Levels of Mastery

For each topic, the CorBoK defines the required Bloom's level (see Appendix B) in both the foundation and the two concentrations. Deciding the appropriate Bloom's level has been quite challenging and is likely to change significantly between version 0.5 and 1.0 of GRCSE based on reviewer feedback and additional author deliberations. The following brief definitions (please see Appendix C, Table 20 for the full definitions) for each of the six Bloom's levels in the cognitive domain were provided:

- **Knowledge –** The ability to remember previously learned material.
- **Comprehension –** The ability to understand information and grasp the meaning of material presented.
- **Application** – **The a**bility to use learned material in new and concrete situations.
- **Analysis –** The ability to decompose learned material into constituent parts in order to understand the structure of the whole. This includes seeing patterns, the organization of parts, recognition of hidden meanings, and the obvious identification of parts.
- **Synthesis –** The ability to put parts together to form a new whole. This involves the use of existing ideas to create new ones, generalizing from facts, relating knowledge from several areas, as well as predicting and drawing conclusions.
- **Evaluation:** The ability to pass judgment on the value of material within a given context or purpose. This involves making comparisons and discriminating between ideas, assessing the value of theories, making choices based on reasoned arguments, verifying the value of evidence, and recognizing subjectivity.

The GRCSE authors went through several iterations to produce the recommendations shown here, collecting inputs from several sources and iterating in an author workshop and through emails and other forms of collaboration. Significant reviewer feedback is sought to help mature these recommendations.

There is one table for each of Parts 2 through 6 of the SEBoK. In Table 4 through Table 8, recommendations are made for every topic in every knowledge area for the foundation. If a concentration has a higher Bloom level than the foundation, the level is also recorded in the table; otherwise, the table cell is left blank.

Table 4. CorBoK Bloom's Levels for Part 2: System Topics (table developed for BKCASE).

Knowledge Area	Topic	Foundation	SEM	SDD
Systems Overview	What is a System?	Comprehension		
	System Context	Comprehension		
	Overview of System Science	Comprehension		
	Systems Thinking	Comprehension		
	Overview of System Concepts	Comprehension		

Knowledge Area	Topic	Foundation	SEM	SDD
System Concepts	System Context	Comprehension		
	Complexity	Comprehension		
	Emergence	Comprehension		
Types of Systems	Classifications of Systems	Knowledge		
	Groupings of Systems	Knowledge		
	The Product View of Engineered Systems	Comprehension		
	The Service View of Engineered Systems	Comprehension		
	The Enterprise View of Engineered Systems	Comprehension		
Representing Systems with Models	What is a Model?	Comprehension		
	Why Model?	Comprehension		
	Types of Models	Comprehension		
	System Modeling Concepts	Comprehension		
	Modeling Standards	Comprehension		
Systems Approach	Overview of the Systems Approach	Comprehension		
	Exploring a Problem or Opportunity	Comprehension		
	Systems Analysis Approach	Comprehension		
	Synthesis of a System	Comprehension		
	Proving a System	Comprehension		
	Owning and Making Use of a System	Comprehension		
	Applying the Systems Approach	Comprehension		
Systems Challenges	Complex System Challenges	Comprehension		
	Dynamically Changing Systems	Comprehension		

Table 5. CorBoK Bloom's Levels for Part 3: SE and Management (table developed for BKCASE).

Knowledge Area	Topic	Foundation	SEM	SDD
Life Cycle Models	Life Cycle Characteristics	Application		
	System Life Cycle Process Drivers and Choices	Application		
	Representing System Life Cycle Process Models: Iterative	Application		
	Representative System Life Cycle Process Models: Vee	Application		
	Integration of Process and Product Models	Application		
System Definition	Fundamentals of System Definition	Application		Analysis
	Mission Analysis and Stakeholders Requirements	Application		Analysis
	System Requirements	Application		Analysis
	Architectural Design	Application		Analysis
	System Analysis	Application		Analysis

Knowledge Area	Topic	Foundation	SEM	SDD
System Realization	System Implementation	Application		Analysis
	System Integration	Application		Analysis
	System Verification and Validation	Application		Analysis
System Deployment and Use	System Deployment	Comprehension		Application
	Operation of the System	Comprehension		Application
	System Maintenance	Comprehension		Application
	Logistics	Comprehension		Application
SE Management	Planning	Comprehension	Analysis	
	Assessment and Control	Comprehension	Analysis	
	Risk Management	Comprehension	Analysis	
	Measurement	Comprehension	Analysis	
	Decision Management	Comprehension	Analysis	
	Configuration Management	Comprehension	Analysis	
	Information Management	Comprehension	Analysis	
	Quality Management	Comprehension	Analysis	
Product and Service Life Management	Service Life Extension	Comprehension	Analysis	Application
	Capability Updates, Upgrades, and Modernization	Comprehension	Analysis	Application
	Disposal and Retirement	Comprehension	Analysis	Application
SE Standards	Relevant Standards for SE	Comprehension		
	Alignment and Comparison of the SE Standards	Comprehension		
	Application of SE Standards	Comprehension		

Table 6. CorBoK Bloom's Levels for Part 4: Applications of SE (table developed for BKCASE).

Knowledge Area	Topic	Foundation	SEM	SDD
Product SE	Product of SE Background	Knowledge		
	The Product as a System	Knowledge		
	Related Business Activities			
	Product of SE – Key Aspects			
	Product SE Special Activities	Knowledge		
Service SE	Service Systems Background	Comprehension		
	Fundamentals of Services	Comprehension		
	Properties of Services	Knowledge		
	Scope of Service SE	Knowledge		
	Value of Service SE	Knowledge		
	Service SE Stages	Knowledge		
Enterprise	Enterprise SE Background	Knowledge	Comprehension	

Knowledge Area	Topic	Foundation	SEM	SDD
Systems Engineering	The Enterprise as a System	Knowledge	Comprehension	
	Related Business Activities	Knowledge	Comprehension	
	Enterprise SE Key Concepts	Knowledge	Comprehension	
	Enterprise SE Process Activities	Knowledge	Comprehension	
	Enterprise Capability Management	Knowledge	Comprehension	
Systems of Systems (SoS)	Architecting Approaches for Systems of Systems	Knowledge		Comprehension
	Socio-Technical Features of Systems of Systems	Knowledge		Comprehension
	Capability Engineering	Knowledge		Comprehension

Table 7. CorBoK Bloom's Levels for Part 5: Enabling SE (table developed for BKCASE).

Knowledge Area	Topic	Foundation	SEM	SDD
SE Organizational Strategy	Organizational Purpose	Knowledge	Comprehension	
	Value Proposition for SE	Knowledge	Comprehension	
	SE Governance	Knowledge	Comprehension	
Enabling Businesses and Enterprises to Perform SE	Deciding on Desired SE capabilities within Business and Enterprises	Knowledge	Comprehension	
	Organizing Business and Enterprises to Perform SE	Knowledge	Comprehension	
	Assessing SE Performance of Business and Enterprises	Knowledge	Comprehension	
	Developing SE Capabilities within Businesses and Enterprises	Knowledge	Comprehension	
	Culture	Knowledge	Comprehension	
Enabling Teams to Perform SE	Determining Needed SE Capabilities in Teams	Knowledge	Comprehension	
	Organizing Teams to Perform SE	Knowledge	Comprehension	
	Assessing SE Performance of Teams	Knowledge	Comprehension	
	Developing SE Capabilities within Teams	Knowledge	Comprehension	
	Team Dynamics	Application		
Enabling Individuals to	Roles and Competencies	Knowledge	Comprehension	
	Assessing Individuals	Knowledge	Comprehension	

Knowledge Area	Topic	Foundation	SEM	SDD
Perform System Engineering	Developing Individuals	Knowledge	Comprehension	
	Ethical Behavior	Application		

Table 8. CorBoK Bloom's Levels for Part 6: Related Disciplines (table developed for BKCASE).

Knowledge Area	Topic	Foundation	SEM	SDD
Introduction	SE Related Disciplines and Hardware Engineering	Comprehension		Application
SE and Software Engineering (SwE)	The Nature of Software	Comprehension		Application
	An Overview of the SWEBOK Guide	Comprehension		Application
	Software Engineering and SE: Similarities and Differences	Comprehension		Application
Systems and Project Management	An Overview of Project Management	Comprehension	Application	
	SE and Project Management: Similarities and Differences	Comprehension	Application	
SE and Specialty Engineering	Integration of Specialty Engineering	Comprehension		Application
	Reliability, Availability, and Maintainability	Comprehension		Application
	Human System Integration	Comprehension		Application
	Safety Engineering	Comprehension	Application	Application
	Security Engineering	Comprehension	Application	Application
	System Assurance	Comprehension	Application	Application
	Environment	Comprehension	Application	Application
	Resilience Engineering	Comprehension		Application
	Manufacturability and Producibility	Comprehension		Application
	Affordability	Comprehension		Application
	Disposal and Retirement	Comprehension		Application

Some general observations about the recommendations:

- Most topics are to be covered at the Knowledge or Comprehension level. This is typically achieved through readings and lectures with simple exercises. Relatively little classroom time is needed to achieve these levels.
- There are 15 topics at the Application level in the Foundation and none at the Analysis level.

- SEM elevates 6 topics from the Comprehension level up to the Application level and elevates 11 to the Analysis level; hence, SEM requires 21 topics be covered at the Application level and 11 at the Analysis level
- SDD elevates 18 topics from the Comprehension level up to the Application level and elevates 8 topics from the Application level to the Analysis level; hence, SDD requires 33 topics to be covered at the Application level and 11 at the Analysis level
- The SEBoK includes recommended primary and additional readings for every topic. These recommendations become natural sources for textbooks and papers to be included in courses that cover the CorBoK topics.

All of these topics must be taught with about 240 classroom contact hours between students and faculty. A somewhat simplistic model assumes 6 contact hours (typically two weeks of lecture) plus significant homework and projects to achieve one topic at Application level and 9 contact hours (typically three weeks of lecture) plus significant homework and projects to achieve one topic at Analysis level. Under those assumptions, an SEM student would need 225 contact hours to cover just the topics recommended for the Application and Analysis levels, leaving 15 contact hours for all the other topics recommended at the Knowledge and Comprehension level. The SDD student would need 297 hours just to cover the topics recommended for the Application and Analysis levels. However, the achievement of Application and Analysis levels requires addressing multiple topics at once. For example, the five topics related to Life Cycle Models (see Table 5) would not be presented as five separate topics (characteristics, drivers, models, etc.) but rather as one coherent set of topics, allowing synergies between the topics. Therefore, to achieve the Application and Analysis levels, topics will naturally be grouped, and the SEBoK knowledge areas group the topic areas in a way that can be leveraged for this effort. Because of synergies provided by topic grouping, contact hours guidance in GRCSE is provided at the knowledge area, rather than topic level. In addition, GRCSE supports different curriculum models that may have more or less contact hours than the baseline 240 hours, and so percentages are used for guidance. This is the model adopted by the GRCSE team and presented in the next section.

6.4 Defining CorBoK Touchpoint Hours (%)

The final step taken by the GRCSE team for version 0.5 was to provide guidance on the contact hours needed across a program to cover each identified topic to the level of detail necessary for the student to achieve the identified Bloom's level. Using the synergy of topics model and percentages (for flexibility in adoption) for the contact time, the first step was to distribute contact time at the parts level. To this end, the GRCSE team, over a series of meetings beginning with the London 2011 BKCASE workshop, allotted the contact time distribution across the parts of the SEBoK being covered by the CorBoK. This distribution is different depending on the concentration selected (differences in % of effort reflect differences in the minimum Bloom's levels for each concentration, combined with Bloom's levels for the foundation topics); therefore two sets of distributions were defined as shown in Table 9.

Table 9. Distribution of CorBok by SEBoK Parts Using Percentages (table developed for BKCASE).

SEBoK Part	Foundation/ SEM %	Foundation/ SDD %
Part 2: Systems	12%	12%
Part 3: Systems Engineering and Management	56%	56%
Part 4: Applications of Systems Engineering	10%	10%
Part 5: Enabling Systems Engineering	10%	4%
Part 6: Related Disciplines	12%	18%
Total Distribution %	100%	100%

Recall that for any master's degree in SE, learning the *CorBoK* occupies up to half the student's time. This means that the total (100%) time for each concentration shown in the table represents no more than half (50%) of the time a student spends in a master's program overall.

As a final step, the GRCSE team distributed the percentages allotted for each part, across the knowledge areas contained in that part, in a consistent manner based on the Bloom's level required for the majority of the topics in that knowledge area. For example, the GRCSE team had previously determined that the majority (56%) of the contact hours for any SE graduate needed to be focused on the topics in Part 3. This percentage was consistently applied such that the most time was focused in those knowledge areas that contained sets of topics to be attained at the Analysis Bloom's level (12%), next, at the Analysis level (8%), and the lowest amount of time for those at the Comprehension level (4%). For a 480-hour program of which the CorBoK is 50% or 240 hours, this roughly translates to 30 contact hours for Analysis, 20 for Application, and 10 for Comprehension.

In contrast, the GRCSE team determined that just over one-tenth (12%) of the contact hours were needed to attain the Bloom's levels provided for Part 2 (Systems). In this case the Bloom's level to be attained were primarily at the Comprehension level and the contact hours were distributed evenly across the six knowledge areas (2% each). For the same 480-hour program, this translates to about 5 contact hours for each knowledge area.

The hours/percentages per topic are the purview of the institution with the guidance that the effort be distributed as needed to the topics within the knowledge areas to achieve the minimum Bloom's levels.

Table 10. Distribution of CorBok by SEBoK Knowledge Areas Using Percentages (table developed for BKCASE).

Part	Knowledge Area	Foundation/ SEM %	Foundation/ SDD %
Part 2: Systems	Systems Overview	2%	2%
	System Concepts	2%	2%
	Types of Systems	2%	2%
	Representing Systems with Models	2%	2%
	Systems Approach	2%	2%
	Systems Challenges	2%	2%
Part 3: Systems Engineering and Management	Life Cycle Models	8%	8%
	System Definition	8%	12%
	System Realization	8%	12%
	System Deployment and Use	4%	8%
	Systems Engineering Management	12%	4%
	Product and Service Life Management	12%	8%
	Systems Engineering Standards	4%	4%
Part 4: Applications of Systems Engineering	Product Systems Engineering	2%	2%
	Service Systems Engineering	2%	2%
	Enterprise Systems Engineering	4%	2%
	Systems of Systems (SoS)	2%	4%
Part 5: Enabling Systems Engineering	Systems Engineering Organizational Strategy	3%	1%
	Enabling Businesses and Enterprises to Perform Systems Engineering	3%	1%
	Enabling Teams to Perform Systems Engineering	2%	1%
	Enabling Individuals to Perform Systems Engineering	2%	1%
Part 6: Related Disciplines	Introduction	2%	3%
	Systems Engineering and Software Engineering	2%	3%
	Systems Engineering and Project Management	3%	2%
	Systems Engineering and Specialty Engineering	5%	10%

Much of the time spent in moving from version 0.5 to version 1.0 of GRCSE will be spent validating, refining, and adjusting these recommendations, especially the hours recommended to address topics at the Application and Analysis levels. This will undoubtedly lead to changes in the topics recommended at the Application and Analytic levels. *Review comments that focus on these recommendations are strongly sought*.

This page intentionally left blank.

7. Implementation

7.1 Introduction

The elements of the *Graduate Reference Curriculum for Systems Engineering* (GRCSE) consist of the objectives, outcomes, entrance expectations, architecture, and the CorBok. This chapter discusses how an institution would incorporate those elements into its programs. The next chapter will discuss how an institution would assess its programs considering these elements.

This chapter complements Appendix F, Competency-Based Curriculum Development, and Appendix G, Use Cases.

7.2 Understanding initial conditions

Each institution begins an implementation of GRCSE from a different starting point. Some may have many of the elements of GRCSE recommendations already in place, while others may have none. The quality of the existing elements may be satisfactory, or not. The first step for every institution, then, is a self-analysis of the current program or programs. What is the existing curriculum? What are/were the design considerations for the curriculum? Who are the stakeholders and key constituencies, and what are their needs? What existing documentation exists?

The survey results discussed in Appendix B suggest that most graduate programs do not have formal objectives stated. However, few programs are developed in isolation from external stakeholders, particularly potential employers. Understanding how the external stakeholder needs have been elicited and compiled is an important part of self-analysis.

Squires (Squires et al. 2011) provides a useful table for assessing program maturity with respect to GRCSE. It is reprinted below as Table 11. Programs are categorized as initial, emerging, developed, or highly developed.

Table 11. Systems Engineering Program Development Phases Mapped to GRCSE (table developed for BKCASE).

	Initial	Emerging	Developed	Highly Developed
Expected Student Background	Meets none of the GRCSE expectations.	Requires a technical undergraduate degree or two years of experience.	Most students meet the expected GRCSE background.	All students meet the expected GRCSE background.
Objectives	Objectives are not established or do not align with GRCSE.	Objectives have been established and address the level of ability in SE practice.	Objectives address most of the GRCSE objectives.	Program objectives have been shaped by program stakeholders and address all GRCSE objectives.

<table>
<tr><th></th><th>Initial</th><th>Emerging</th><th>Developed</th><th>Highly Developed</th></tr>
<tr><td>Outcomes</td><td>Program outcomes are not established or do not align with GRCSE.</td><td>Outcomes address the core body of knowledge.</td><td>Outcomes address most of the GRCSE outcomes.</td><td>Program outcomes address all GRCSE outcomes.</td></tr>
<tr><td>Preparatory Knowledge</td><td>The program admits students without preparatory knowledge and are not remediated.</td><td>The program provides students with opportunities to master some of the preparatory knowledge.</td><td>The program provides students opportunities to master all of the preparatory knowledge.</td><td>The program requires students to demonstrate mastery of all preparatory knowledge before proceeding in the program.</td></tr>
<tr><td>Core Foundation Knowledge*</td><td rowspan="4">The program does not address the GRCSE knowledge area.</td><td rowspan="4">The program addresses portions of the knowledge area according to GRCSE recommendations.</td><td rowspan="4">Each knowledge area is fully addressed according to GRCSE recommendations.</td><td rowspan="4">The program requires students to demonstrate mastery at the appropriate Bloom’s level for each knowledge area. Each knowledge area is fully addressed according to the GRCSE recommendations.</td></tr>
<tr><td>Core Extension Knowledge*</td></tr>
<tr><td>Domain-Specific Knowledge*</td></tr>
<tr><td>Program-Specific Knowledge*</td></tr>
<tr><td>Capstone Experience</td><td>The program does not offer a capstone experience.</td><td>The program offers, but does not require, a capstone experience.</td><td>The program requires a capstone experience.</td><td>The program requires students to demonstrate their accumulated skills and knowledge in a capstone experience.</td></tr>
<tr><td>Program Assessment</td><td>The program does not have a formal assessment plan.</td><td>The program has an assessment plan.</td><td>The program has an assessment process that includes evidence of resulting program improvement.</td><td>The program has a comprehensive assessment process that includes direct measures and evidence of resulting program improvement.</td></tr>
</table>

*Mapping description should be separately applied to each knowledge area.

7.3 Developing objectives

Program objectives should be developed or updated through consultation with stakeholders. They should be consistent with the mission of the institution, published in a form accessible to the public, and updated periodically. Many programs have industrial advisory boards representing major employers for program graduates, and these boards are one formal mechanism for providing stakeholder input as objectives are developed or updated. Surveys are also used to elicit stakeholder needs.

Undergraduate engineering programs uniformly maintain program objectives to meet accreditation requirements. Most graduate programs do not seek accreditation and so do not bother with formal development of objectives even though they maintain them for their undergraduate programs. GRCSE specifically recommends the discipline of developing and assessing program objectives for SE graduate programs. Most programs can leverage their existing undergraduate mechanisms to develop objectives for their graduate programs. For example, an existing undergraduate advisory board may provide an initial starting point for developing a program advisory board for a graduate program.

7.4 Developing outcomes

Many GRCSE expected outcomes explicitly require tailoring by the program. For example, Concentration, the second outcome requires the program to select a concentration area. The third, fourth, and fifth outcomes, Topic Depth, Application Domain, and Specialty, require program depth and specialization, but the specific areas are left to the discretion of the program based on its program objectives, special competencies, and other factors. Faculty curricular committees are the usual mechanism for this tailoring.

7.5 Addressing the CorBoK

The *Core Body of Knowledge* (CorBoK) lists over a hundred topics extracted from the *Guide to Systems Engineering Body of Knowledge* (SEBoK), and recommends a minimum Bloom's level for each topic, both for the foundation and the concentrations. Attaining these levels is expected to require about half the student contact hours in the curriculum, leaving the other half for program unique content. Much of the local specialization involves treating topics in the CorBoK to obtain a higher attainment than the minimum and the CorBoK tables can be used as a framework for planning that specialization.

The initial analysis of the program may identify gaps in CorBoK coverage. These gaps can be addressed by modifying existing courses, by adding courses, by a redesign of the capstone experience, or by adjusting entrance expectations. Each gap should have a mitigation approach identified for it.

7.6 Setting entrance requirements

In Chapter 4, GRCSE recommends a set of student entrance expectations that should inform a program's entrance requirements; i.e., the actual criteria a program sets for admitting students. The ability and background of the entering student shapes the amount of instruction necessary to meet the GRCSE outcomes. Highly selective entrance requirements can support program quality, but reduce the number of students eligible for a program. Each program must consider its institutional mission, stakeholder needs, accreditation requirements, and market forces when setting its particular entrance requirements.

Relaxing entrance requirements results in either increased preparation work (increasing program length), or reduced levels of attainment of the outcomes.

In particular, the knowledge of computing fundamentals and the work experience entrance expectations should be carefully considered. The absence of computing fundamentals severely impacts the ability of a student to master the software engineering (SwE) material, and a lack of engineering experience affects the team outcomes and much of the expected background for the CorBok.

The domain specialization selected for outcome four, Application Domain, may require additional technical prerequisites for admission.

7.7 Architecting courses and capstones

After objectives, outcomes, and entrance expectations have been set, a program can then consider the courses and capstone experiences that will constitute the curriculum. This requires mapping the outcomes to the entrance prerequisites, courses, and capstone experience.

Design choices include the use of design problems that span courses, use of existing courses versus development of new ones, and the length and type of capstone experience.

The SEBoK includes primary and additional references for each topic. Those references become obvious candidates for textbooks and papers for courses that implement the program.

Programs that relax the entrance expectations for some but not all students may develop a set of preparatory courses to address relevant areas.

7.8 Designing the assessment process

Details on assessment are provided in Chapter 8 and Appendix E. When implementing an assessment process, key decisions include the elements to be assessed, the frequency of assessment, the method of assessment, who will provide evidence, and the process management.

Elements assessed include those items on the right hand of the Vee in Figure 2. There are multiple assessment perspectives, including assessing the design of the elements, the delivery of the elements, and the student mastery of the associated elements. Students who master a curriculum designed with poor outcomes and inappropriate objectives may not be well served, despite their best efforts. The best designed program will suffer if it is poorly delivered.

A two year or three year assessment cycle is recommended. As discussed in Chapter 8, a focused assessment plan that collects evidence judiciously can be implemented without excessively burdening the faculty, students, stakeholders, and staff.

7.9 Continuous Improvement

Assessments are useless if they are not used as part of a continuous improvement plan. Assessments should feed a periodic review and update of the curriculum, providing evidence of where improvements are needed and of the success of earlier efforts. A best practice is the use of a log that documents issues that arise, the source of the concern (be it assessment tool or other input), the response, and the effectiveness of the response (with evidence).

7.10 Examples of Implementation

Version 1.0 of GRCSE will include the implementation experience of early adopters of GRCSE. To be included, contact the authors at bkcase@stevens.edu.

This page intentionally left blank.

8. Assessment

8.1 Introduction

How does a program evaluate whether its educational objectives and desired outcomes are being achieved? It uses the process of identifying, collecting, and analyzing data known as assessment. Schools and universities rely on assessment to

- determine how well outcomes correspond to student and societal needs and how well student activities, products, and performance coincide with the academic community's expectations;
- take a central role in global processes for improving program quality; and
- provide accreditation authorities (including the US ABET, French CTI, UK Engineering Council, and German ASIIN, among others) with information these authorities deem mandatory.

In short, assessment is important in education because without it, needs may not be met, there is no assurance of quality improvement, and accreditation cannot occur.

The *Guide to the Systems Engineering Body of Knowledge* (GRCSE) addresses assessment in two complementary ways:

In Appendix E, GRCSE presents assessment as an iterative four-step management method based on the *plan, do, check, act* (PDCA) approach used for the control and continuous improvement of processes or products. The objective is not to obtain an absolute estimate of a program quality, but rather to measure its relative evolution between iterations. This general framework for assessment defines terms, introduces principal processes, and presents major methods to support assessment. It is generic enough to serve as a guide for those who want an overall introduction to assessment and a reference for developing an assessment methodology that aligns with GRCSE principles.

On the other hand, this chapter describes the application of an assessment framework to graduate programs in SE. Rather than describe a full and rigid use of the methodology, a series of examples is considered to highlight the balance between efficiency and effectiveness in assessment, and elucidate some typical difficulties. The material has been chosen to help academics define and establish assessment processes in harmony with the general principles found in Appendix E.

Assessment covers a broad scope, from the individual student to the academic institution, in between which are courses, program outcomes (what students are expected to know and be able to do by the time of graduation), and program objectives (what graduates are expected to attain within a few years of graduation). For more detail, see Appendix E.1.

Achievement of program outcomes is demonstrated using assessments of course outcomes; achievement of courses outcomes is demonstrated using assessments of individual student performance. So the 'targets' of assessment should not be considered in isolation. Figure 8 below illustrates their interdependence.

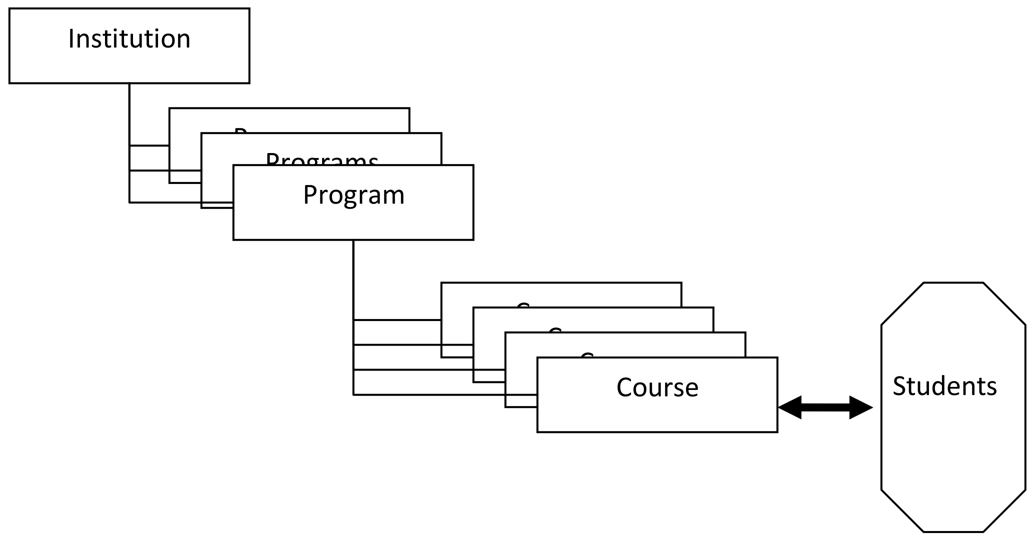

Figure 8. Targets for Assessment (figure developed for BKCASE).

The sections that follow show how assessment is developed by some representative schools and universities that teach SE. The story begins with an assessment of program objectives, then program outcomes, then courses, then students, and concludes with a look at how schools take action based on the results of their assessment process.

8.2 Assessing Program Objectives

Program objectives concern what graduates are expected to attain within a few years of graduation.

Measuring the attainment of objectives directly is difficult because graduate work products are generally not obtainable by the school. Instead, programs may survey the graduates and ask them to forward another survey to their employers for separate submission. The surveys request data about the graduate (e.g., year of graduation and program studied) and the graduate's performance and compare them with program objectives. Locating and then getting good response rates from graduates and their employers can be challenging.

Other methods include the establishment of an industrial advisory board for the curriculum, and placing major employers on the board. The board can provide feedback on the selection of project objectives and how well the program is attaining these objectives.

To illustrate important concepts, an extended example that follows a fictitious school, Simulated University (Sim U), as it establishes objectives and outcomes and associated performance measures for its SE program is presented in this chapter. The school defines where and when the objectives and outcomes will be assessed and what the targets are. Many approaches are possible. The school in the example follows the process described in Appendix E.

To begin the example, Sim U asks graduates to indicate their agreement or disagreement, on an ordinal Likert scale (from one to five), with statements such as

The Sim U Systems Engineering program provided me with an educational foundation that allowed me to [insert a program objective].

Employers are asked similar questions, and whether they agree with the program objectives. Free-text responses are also collected and tabulated.

The data, free-text responses, and board feedback are evaluated by a faculty committee. This is done on a multi-year cycle to reduce data collection, evaluation fatigue, and respondent burnout.

8.3 Assessing Program Outcomes

Program outcome assessment focuses on the competencies and capabilities of students at graduation to determine if the program is meeting its outcomes. Potential improvements are identified and implemented and their effectiveness tracked.

Step 1 – Identify program outcomes

To be effectively assessed, outcomes must be explicitly stated for a curriculum. The survey of graduate SE programs cited in Appendix B showed that only 11 of the 31 responding programs had established program outcomes that described what student capabilities would be upon graduation. Other programs listed objectives or courses taken as outcomes or stated that outcomes were under development.

Continuing the example, Sim U develops the following outcome, based on GRCSE outcome, Topic Depth, as one of the outcomes for its system engineering program:

Step 1.1 –Achieve Bloom's synthesis level of attainment for system analysis.

Outcomes are often associated with performance criteria. Sim U goes on to set four performance criteria for students at its program outcome:

- Develop measures for a trade-off study.
- Perform a quantitative trade-off study that addresses multiple objectives.
- Prepare a project cost estimate that utilizes multiple techniques and incorporates probabilistic bounds.
- Develop a system risk assessment that addresses technical, schedule, cost, and performance risks.

Good measures are at least ordinal and based on data that is straightforward to obtain. Sim U decides to measure each indicator by an ordinal numerical score on a common one-to-five scale and sets a target for each performance measure of a mean score of four or greater. Other targets may be tailored to a program; for example, that 80% of student work is assessed as a four or better.

Step 2 – Define assessment strategy and methods

Having decided *what* to measure, the school now asks *how* to measure, *where* in the curriculum to measure, *when* to measure, *who* will measure, and what the targets are for the measures.

These answers take the form of an assessment strategy and methods with which to realize that strategy.

Step 2.1 – Define assessment methods

Sim U chooses to perform assessment by direct methods only:

- Assess student theses or capstone projects

  Student theses and capstone projects can be assessed against all outcomes and performance measures. While few are likely to show evidence of all outcomes, each should shed light on some outcomes.

- Assess student projects (portfolio)

  Student projects or specific examination questions can be measured against criteria for student accomplishment. Sim U notes the courses that covered each performance criteria and selects one or more courses where suitable student work can be assessed at the end of the term.

Sim U chooses not to involve employers of graduates, which would have given an interesting external point-of-view. This could have been done through employer surveys or the establishment of an advisory board.

Step 2.2 – Define the life cycle of assessment

It is not necessary to assess every student in every course offering. Sim U decides to assess coursework every two years in the fall quarter and randomly sample from the population of student work for that quarter. It also decides to assess every student thesis or capstone project year-round.

Step 2.3 – Define the organization for assessment

Sim U establishes a faculty committee to perform the assessments, thus avoiding asking course directors and instructors to assess themselves. Thesis advisors, capstone advisors, and the department chair each prepare an evaluation of the thesis or capstone**.**

Step 3 – Implement changes

As data is collected in each assessment cycle, Sim U evaluates whether or not action is needed to improve the program. That action can be a change in the instructional approach, a change in the assessment process, or a change in the target. Sim U documents both the evaluation and the actions taken.

The example in Table 12 below shows that capstone and portfolio assessment indicate that the program is meeting the outcome target for each performance measure. Accordingly, Sim U decides that it needs to take no action on this particular outcome. Other outcomes are documented similarly, and show Sim U where action for improvement needs to be taken.

Table 12. Example of a Single Page Assessment Plan for an Outcome (table developed for BKCASE).

Student Outcome: *Achieve a Bloom's synthesis level of attainment for system analysis.*

Performance Criteria	Courses with coverage	Method(s) of assessment	Where data are collected	Length of assessment schedule	Year / semester of data collection	Target for performance
Develop measures for a trade-off study	SE3XXX, SE3XXY, SE4XXX	Capstone and portfolio evaluation	Graduation, SE4XXX	Two years	Wtr 2012	Mean > 4.0
Perform a quantitative trade-off study that addresses multiple objectives	SE3XXX, SE3XXY, SE4XXZ	Capstone and portfolio evaluation	Graduation, SE4XXZ	Two years	Wtr 2012	Mean > 4.0
Prepare a project cost estimate that utilizes multiple techniques and incorporates probabilistic bounds.	SE3XXX, SE3zXY, SE4XAX	Capstone and portfolio evaluation	Graduation, SE4XAX	Two years	Wtr 2012	Mean > 4.0
Develop a system risk assessment that addresses technical, schedule, cost, and performance risks.	SE3XXX, SE3XXY, SE4XXW	Capstone and portfolio evaluation	Graduation, SE4XXW	Two years	Wtr 2012	Mean > 4.0

8.4 Accessing Course Delivery

Courses have learning goals and teaching methods. Assessing a course brings these to light and allows colleagues to collectively examine practices that may otherwise remain private. The objective is to improve learning, not to evaluate the teachers or students.

The next example comes from a "non-specialized engineers" program with SE related outcomes taught at a university focused on aeronautics-oriented training programs and activities. Students and trainees are educated to become engineers, pilots, and air traffic controllers, especially in the air transport sector.

Step 1 – Identify course outcomes

The school defines expected outcomes for each course in the syllabus. These deploy phrases such as "Upon successfully completing the course, students are able to . . ." Program outcomes are mapped to course outcomes, as illustrated in Table 13 below.

Table 13. Extract of the Syllabus (table developed for BKCASE).

Program outcomes	Course outcomes	Course reference
At the end of the program, students should be able to define and apply a Verification and Validation (V&V) strategy.	Upon successfully completing the course, students understand basic definitions and objectives of verification and validation processes	SE3: Introduction to V&V
	Upon successfully completing the course, students know the four basic means for supporting verification and validation processes: review, demonstration, analysis and test.	SE3: Introduction to V&V
	Upon successfully completing the course, students are able to set up a verification and validation plan, including: • the definition and justification of a V&V strategy; • the identification of V&V activities life cycle; • the specification of the environment; and • the definition of the organization.	SE3: Introduction to V&V
	Upon successfully completing the course, students should be able to have an in-depth knowledge in model based system engineering approach, including: • model checking approach and • proof model approach.	SE3-1: Introduction to formal methods
	Upon successfully completing the course, students should be able to have an in-depth knowledge in test techniques, including: • white-box and black-box coverage and • test tools and environment.	SE3-2: Introduction to testing

Step 2 – Define assessment strategy and methods

The school decides to split the evaluation of each course into measures of student achievement and course delivery effectiveness.

The school considers the learning level reached by students during a course to be a good indicator of course health. Through a black-box assessment (where the internal workings of the system are assumed unknown), teachers can assess their teaching methods based on the individual evaluation of their students. An example of a method for mapping course objectives to course assessment methods is shown in Table 14. How the school performs student assessment is covered in the next section.

Table 14. Direct mapping is done between course outcomes and assessment methods (table developed for BKCASE).

Course reference	Course outcomes	Student assessment method	Other assessment method
SE3: Introduction to V&V	Upon successfully completing the course, students will be able to understand basics definitions and objectives of verification and validation processes	Multiple choice examination questions	Survey for students Survey for teacher
SE3: Introduction to V&V	Upon successfully completing the course, students will be able to know the four basic means for supporting verification and validation processes: review, demonstration, analysis, and test.	Short answer examination questions	Survey for students Survey for teacher
SE3: Introduction to V&V	Upon successfully completing the course, students will be able to set-up a verification and validation plan including: • the definition and justification of a V&V strategy; • the identification of V&V activities life cycle; • the specification of the environment; and • the definition of the organization.	Course project: develop a V&V plan	Survey for students Survey for teacher
SE3-1: Introduction to formal methods	Upon successfully completing the course, students should be able to have an in-depth knowledge in model based system engineering approach, including: • model checking approach and • proof model approach.	Course project: MBSE	Survey for students Survey for teacher
SE3-2: Introduction to testing	Upon successfully completing the course, students should be able to have an in-depth knowledge in test techniques, including: • white-box and black box coverage and • test tools and environment.	Essay examination question	Survey for students Survey for teacher

The school also values the way in which the course is performed as an indicator of course health. In the white-box assessment at the end of each course, students and teachers are required to complete course satisfaction surveys, which are anonymous, private, and accessible online to authorized users.

For a dedicated list of topics, each teacher involved in a course marks his or her level of satisfaction with the course, as illustrated in Table 15.

Table 15. Example of a Teacher Online Survey (table developed for BKCASE).

Course : XXXXX		***Satisfaction***				
Topics	***Low***	**1/4**	**2/4**	**3/4**	**4/4**	***Maximum***
Definition of educational objectives	Insufficient	○	○	○	○	Sufficient
Suitability of course for the target population	Incorrect	○	○	○	○	Correct
Class size	Inadequate	○	○	○	○	Suitable
Pre-requisites for students to follow the course	Insufficient	●	○	○	○	Sufficient
Achievements (assimilation by students)	Poor	○	○	○	○	High

Finally, in free-text fields, the survey requests comments on the assessment itself and general suggestions.

The student survey forms are similar to those for teachers, but include additional topics.

The data collected is analyzed by the academics responsible for the course. A count of responses is tallied for each possible response for each topic. Mean and deviation values are calculated.[8]

Each topic with a summary mark less than 2.5 is carefully studied and corrective actions are defined. Moreover, when this kind of survey is repeated over the years on the same course, it becomes possible to follow trends in quality indicators, and to review the effectiveness of corrective actions. Table 16 below shows the kind of results obtained.

[8] The nominal scale used makes the statistics returned mathematically invalid, but they remain useful as an overall summary of opinion.

Table 16. Example of Student Online Survey Results (table developed for BKCASE).

Course: xxxxx	Teachers: xxxxx				105 forms	
Topic	**1/4**	**2/4**	**3/4**	**4/4**	**mean / 4**	**deviation**
Course						
Knowledge of the course objectives	1	13	58	32	**3.2**	0.7
Suitability course/objectives	2	12	56	34	**3.2**	0.7
Ease of assimilation	4	30	40	30	**2.9**	0.9
Course plan	1	9	42	52	**3.4**	0.7
Relationship course material / course	0	5	33	66	**3.6**	0.6
Course material (subject)	2	16	48	38	**3.2**	0.8
Course material (form)	4	26	57	17	**2.8**	0.7
Rhythm during the course	4	29	41	30	**2.9**	0.9
Number of sessions	1	5	41	57	**3.5**	0.6
Teacher availability	1	5	60	38	**3.3**	0.6
Project						
Number of sessions	19	19	34	32	**2.8**	1.1
Difficulty of the subject	6	6	27	65	**3.5**	0.8
Interest for the subject	8	8	31	57	**3.3**	0.9
Autonomy, initiative let to the student	12	10	37	45	**3.1**	1.0
Logistical support	8	14	46	36	**3.1**	0.9
Tutor availability	18	15	39	32	**2.8**	1.1
Test						
Exam level	0	12	46	46	**3.3**	0.7
Correlation with lessons	0	2	46	56	**3.5**	0.5
Exam correction	54	9	29	12	**2.0**	1.1
Appraisal						
General impression	2	16	48	38	**3.2**	0.8
Personal work	12	48	36	8	**2.4**	0.8
Interest in this course	10	29	38	27	**2.8**	0.9
Linking with other courses	1	14	52	37	**3.2**	0.7
Global hourly volume	2	12	46	44	**3.3**	0.7
Course rhythm	4	26	37	37	**3.0**	0.9

8.5 Student Performance Assessment

The student is evaluated throughout the assessment process. Individual student assessment contributes to:

- the establishment of the student's relative position in a class;
- the final assessment of the student, leading to graduation; and
- course evaluation, since the information about student achievement helps determine the level of attainment of course outcomes.

The aeronautical program example from the previous section continues with an account of the student assessment process. Here, student assessment functions as a part of course evaluation. Recall that the faculty has to divide the course evaluation into a black box part to consider student assessment results, and a white box part (where knowledge of the course mechanics is included) to consider how the course was taught. We now describe the black-box exercise, the objective of which is to obtain information that can help improve course performance.

Step 1 – Identify outcomes

The university identifies numerous outcomes for the students. These are inherited from course and program outcomes.

Step 2 – Define assessment strategy and methods

The school decides to choose assessment methods according to the Bloom level associated with each outcome:

- Multiple choice examinations for Bloom "Knowledge" level. Each student has 40 minutes to answer 40 questions.
- Problem solving for Bloom "Comprehension" level. In a written test, each student is asked to analyze and propose a solution to a case study. This problem-solving examination lasts two hours.

Table 17. Possible Methods for Assessing Bloom's Level Achievement (table developed for BKCASE).

Program outcomes	Course outcomes	Bloom level	Student assessment method
At the end of the program, students should be able to define and apply a Verification and Validation (V&V) strategy. (Bloom level 3)	Upon successfully completing the course, students understand basic definitions and objectives of verification and validation processes.	Knowledge	Multiple choice examination
	Upon successfully completing the course, students know the four basic means for supporting verification and validation processes: review, demonstration, analysis and test.	Knowledge	Multiple choice examination
	Upon successfully completing the course, students are able to set up and apply a verification and validation plan, including: • the definition and justification of a V&V strategy; • the identification of V&V activities life cycle; • the specification of the	Application	Capstone (team) project, including: - oral presentation - product creation - written reports

Program outcomes	Course outcomes	Bloom level	Student assessment method
	environment; and • the definition of the organization.		
	Upon successfully completing the course, students possess in-depth knowledge in the model-based SE approach, including: • model checking approach and • proof model approach.	Comprehension	- Multiple choice examination - Problem solving
	Upon successfully completing the course, students possess in-depth knowledge in test techniques, including: • white-box and black box coverage and • test tools and environment.	Comprehension	- Problem solving

- A capstone project for Bloom "Application" level. The project is conducted by a team whose oral presentation, product creation, and reports are assessed.

  The capstone project lasts one full semester. Students are organized into teams of four. They are evaluated on the basis of the assessment of their organization during the project (project definition and follow-up), their production in terms of the final product and associated documents (technical requirement document, design document, verification and validation plan, and result), and their final presentation of the project (final report and oral speech).

  The university is aware that in capstone project teams, a poor performer can hide behind the success of others. Accordingly, the university chooses capstone assessment methods that mitigate this potential problem:

- All team members receive the same evaluation on the final report.
- At the final oral presentation, students who desire an individual assessment are required to perform a part of the presentation and to answer questions about another student's part of the presentation.

8.6 Acting on Assessment Results

Program leaders must be prepared to act on the information gained from assessment. This requires a disciplined process and a strong commitment to program improvement to overcome institutional inertia, similar to 'Vee' development in SE.

Schools implementing changes to their processes following an assessment tend to follow a three-step path (Goda 2010):

1. First, modify the assessment process. It often takes two complete cycles to actually get a sound and sustainable process in place.

2. Second, make major curriculum changes, such as creating new courses.

3. Finally, make minor changes to improve the program. For example, if the results of the "functioning on a team" outcome fall short of expectations, the school might introduce a team programming project in a foundational programming course where all work had previously been done individually. Or, they might dedicate a day of class to instructing the students on building effective teams.

A school's progress along this path is an indication of the maturity of their assessment process.

For more on assessment practices and process steps, including information about acting on the results to implement improvements, see Appendix E.4, Step 6.

9. Evolution

The *Graduate Reference Curriculum for Systems Engineering (GRCSE)*, Version 0.5, is GRCSE's second development release and is subject to worldwide review through March 15, 2012. It is based on Version 0.5 of the *Guide to the Systems Engineering Body of Knowledge* (SEBoK), which was released in September 2011, and on Version 0.25 of GRCSE, which was released in December 2010. The *Body of Knowledge and Curriculum to Advance Systems Engineering* (BKCASE) project, led by Stevens Institute of Technology and the Naval Postgraduate School, is responsible for developing both GRCSE and the SEBoK. BKCASE plans one more interim release of the SEBoK in spring of 2012, eventually culminating in Version 1.0 in September. No additional interim releases of GRCSE are planned. GRCSE 1.0 will be published in December 2012, and will take into account review comments, additional author perspectives, and feedback from audiences at conferences, workshops, and other venues where GRCSE will be presented in that year.

Looking ahead, besides responding to review comments, GRCSE 1.0 will include implementation examples from early adopters, a richer and more detailed *Core Body of Knowledge* (CorBoK), a more mature discussion of use cases in Appendix G, and fuller traceability between GRCSE elements.

The BKCASE project was formed to develop the SEBoK and GRCSE, but not to be their long-term stewards. The original plan, still in effect, was to turn both products over to the Institute of Electrical and Electronics Engineers (IEEE) and the International Council of Systems Engineering (INCOSE) for joint promulgation, maintenance, and evolution. Both the SEBoK and GRCSE need the vantage point and long-term stable funding that only major professional societies can offer if they are to have widespread impact. INCOSE (the most prominent global professional society dedicated to SE) and the IEEE (the largest and most prominent global professional engineering society) are the ideal organizations to become co-stewards of the SEBoK and GRCSE projects.

INCOSE's interest in becoming a co-steward with IEEE has been strong since the project began. The majority of the BKCASE authors are INCOSE members, many in leadership positions. Their Board of Directors has repeatedly shown high interest in the BKCASE project and funds three authors to contribute to the writing as well as to represent INCOSE. The Board has passed multiple resolutions expressing their desire to become stewards of both the SEBoK and GRCSE and to authorize investigation of what resources are necessary for that stewardship to be successful.

From the beginning of the project, the IEEE also showed strong interest in becoming a co-steward; however, because the IEEE is much larger than INCOSE, they needed to identify the specific organizational unit to work with BKCASE and INCOSE. The IEEE Computer Society has become the primary interface to BKCASE and INCOSE with support from the IEEE Systems Council. The leadership of the IEEE Computer Society has representatives on the author team contributing to the writing and representing the Society.

A committee of six people formed in the summer of 2011 to draft a stewardship agreement between Stevens Institute of Technology, INCOSE, and the IEEE Computer Society. That agreement is still evolving with the expectation that all parties will sign it by the summer of 2012. The agreement will cover how the products will be disseminated and supported; the periodicity for minor and major revisions, as well as who can authorize those revisions; and will confirm that all revisions will remain free for worldwide use.

Glossary

List of Acronyms

Acronym	Definition
ACM	Association for computing machinery.
BKCASE	Body of Knowledge and Curriculum to Advance Systems Engineering
CorBoK	Core Body of Knowledge
CSEP	Certified SE Professional (used by INCOSE)
DoD	U.S. Department of Defense
GRCSE™	Graduate Reference Curriculum for Systems Engineering
GRE	Graduate Record Exam
GSwE2009	Graduate Software Engineering 2009
GSwERC	Graduate Software Engineering Reference Curriculum project
IEEE	Institute for Electrical and Electronics Engineers
IELTS	International English Language Testing System
INCOSE	International Council on Systems Engineering
iSSEC	Integrated Systems and Software Engineering Curriculum
KA	Knowledge Area
KSA	Knowledge, Skills, Abilities
OSD	Office of the Secretary of Defense (DoD)
NDIA	U.S. National Defense Industrial Association
PMBOK	Project Management Body of Knowledge
SDD	System Development or Design
SE	Systems Engineering
SEBoK	Systems Engineering Body of Knowledge
SEM	Systems Engineering Management
SwE	Software Engineering
SWEBOK	Software Engineering Body of Knowledge
TM	Technical Management
TOEFL	Test of English as a Foreign Language

Terminology

accreditation—Processes governed by organizations specifically responsible for accreditation in each jurisdiction. Each accreditation organization has its own governance processes for its accreditation activities (USDOE 2011).

admission requirement—The minimum standard an individual must meet in order to enter an academic program. These requirements are generally mandatory, and waivers require justification. Admission requirements are not specified in GRCSE. (See also **entrance expectations**)

affective domain—Unit of Bloom's Taxonomy concerned with feelings, appreciation, and valuation responses to the content which is learned. (Schmidt 1975, 36-49)

application domain—A specific field of application in systems engineering, generally related to a specific discipline (e.g. defense, aerospace, finance, medical, etc.).

architecture—See **curriculum architecture**

assessment—Processes that identify, collect, analyze, and report data that can be used to evaluate achievement, and the process of reviewing the results of data collection and analysis and making a determination of the value of findings and action(s) to be taken. (ABET 2010, 1)

bachelor's degree—An undergraduate degree, generally consisting of a three- or four-year course of study, which builds on secondary education.

Bloom's Taxonomy— A categorization of educational outcomes divided into the cognitive, affective and psychomotor domains. The cognitive domain concerns the intellectual aspect of learning. The affective domain concerns valuing things and development of value structures. The psychomotor domain concerns the development of physical skills to perform tasks. GRCSE uses the cognitive domain extensively and the affective domain to a relatively small extent. Appendix D provides a detailed discussion (Bloom 1994).

capstone experience— A project that is conducted by a team whose oral presentation, product creation, and reports are assessed.

cognitive domain—The cognitive domain of Bloom's taxonomy of educational outcomes concerns the intellectual aspect of learning. GRCSE uses the cognitive domain extensively. The taxonomy divides the cognitivie domain into six levels of achievement: Knowledge (K), Comprehension (C), Application (AP), Analysis (AN), Synthesis (SYN), and Evaluation (EV). These levels are used to describe the depth to which curricula should cover specific topics in order to develop appropriate abilities in the student. Appendix D provides a detailed discussion (Bloom et. al. 1956).

competency—A set of related observable behaviors reflecting knowledge, skills, and personal characteristics that distinguish superior performance, which provides the organization with sustainable competitive advantage (Pyster and Olwell, et. al. 2011) (See also Appendix F).

competency assessment—The process of identifying and assessing defined competencies among a group of employees, typically by department, job category, or hierarchical structure.

component—(1) An entity with discrete structure, such as an assembly or software module, within a system considered at a particular level of analysis (ISO/IEC 15026 1998); (2) One of the parts that make up a system (IEEE 829 2008); (3) set of functional services in the software, which, when implemented, represents a well-defined set of functions and is distinguishable by a unique name (ISO/IEC 29881. 2008).

concentration—The set of CorBoK knowledge which should be known by a subset of all SE graduate students, depending on the focus of study. Each student should master at least one concentration.

core body of knowledge (CorBoK)—The recommended knowledge areas and topics that should be contained within a systems engineering master's degree program. This includes both the core foundation and the core extension. The CorBoK will provide a recommendation as to the appropriate Bloom's level for each knowledge area or topic.

course—A collection of material, exercises, and assessment for which academic credit is awarded, which may be part of a number of programs.

curriculum—All the courses associated with a specific course of study. The curriculum will depend on the level (e.g., graduate or undergraduate) and specificity (i.e., discipline or specialty) of the course of study.

curriculum architecture—The structure and framework used to develop a specific course of study. The GRCSE™ curriculum architecture is discussed in Chapter 6.

degree program—A collection of courses, delivered by an appropriate authority, leading to an academic degree.

development stage—The life cycle stage that includes detailed planning, development, and integration, verification, and validation (IV&V) activities.

discipline – a branch of learning

domain-centric—Term used to describe a SE master's program which is focused on a particular field in which systems engineering is applied. This type of program covers systems engineering fundamentals, but will often include unique vocabulary, processes, and considerations for legal, regulatory, and environmental constraints unique to the domain of study. Described further in Fabrycky (2010). See also **system-centric**.

engineering—(1) The application of scientific knowledge to practical problems, or the creation of useful things. The traditional fields of mechanical engineering, electrical engineering, etc. are included in this definition (Checkland 1999); (2) To (cleverly) arrange for something to happen. (Checkland 1999)

entrance expectations—Knowledge and skills expected students should possess when they enter a SE master's program. These are often prerequisites to the topics they will study.

foundation—The set of CorBoK knowledge which should be known by all SE graduate students, regardless of concentration or institution.

graduate education—A graduate or professional-level degree intended to follow an undergraduate program of study. Within GRCSE, a master's degree in systems engineering is focused on developing knowledge, skills, and abilities (KSA) to meet the current and future challenges of complex systems. In other contexts "master's degree" has other meanings, including a research focused program.

human factor—A physical or cognitive property of an individual or social behavior which is specific to humans and influences functioning of technological systems as well as human-environment equilibriums.

integration—A process that combines system elements to form complete or partial system configurations in order to create a product specified in the system requirements (ISO/IEEE 12207 2008, 1).

knowledge—The information and understanding that an individual must have to perform a task successfully. Knowledge provides the basis for performing a skill.

knowledge area (KA)—Organizational element of the *Systems Engineering Body of Knowledge (SEBoK)*; a knowledge area is the second highest organizational element of the SEBoK (the highest orgainzational element is parts) and consists of one or more topics and subtopics.

knowledge, skills, abilities (KSAs) –a student should learn in order to achieve the expected outcomes upon graduation with master's degree in systems engineering

leveling course— A course designed to allow students who do not meet entrance expectations to enroll in an academic program. In general, these are courses designed to ensure that students have the requisite knowledge, skills, and abilities to succeed in the program. These may also be referred to as a bridging course or preparatory course.

life cycle—(1) The organized collection of activities, relationships, and contracts which apply to a system of interest during its life*;* (2) the evolution of a system, product, service, project, or other human-made entity from conception through retirement; (3) Development (life) cycles start with user needs and end with system decommissioning and disposal. Project cycles contain three aspects: business, budget, and technical. (Mooz, Forsberg, Cotterman 2003, 259)

life cycle stage—The different statuses which occur during the life of a system and the transition between various statuses.

master's degree—A graduate or professional-level degree intended to follow an undergraduate course of study. Within GSwE2009, a master's degree in software engineering is focused on developing knowledge, skills, and abilities to meet the current and future challenges of complex systems that require software in order to operate properly.

natural science—A field which focuses on utilizing scientific method, principles, and rigor to explore, describe, and understand the natural world. The natural sciences include disciplines such as biology, chemistry, physics, astronomy, geology, meteorology, oceanography, etc.

objective—Broad statements that describe the career and professional accomplishments that the program is preparing the graduates to achieve. (ABET Engineering Accreditation Commission 2011).

organization—A person or a group of people and facilities with an arrangement of responsibilities, authorities and relationships (ISO 9000 2000).

outcome—A statement of what students are expected to know and be able to do by the time of graduation. These relate to the skills, knowledge, and behaviors that students acquire as they progress through the program. (ABET Engineering Accreditation Commission 2011).

professional degree –A degree that prepares a student for the practice of a profession, such as law, medicine, or engineering. It is distinguished from a research degree that prepares a student for a career in academia or research.

program—A collection of courses, delivered by an appropriate authority, leading to an academic degree.

reference curriculum—A set of outcomes, entrance expectations, architecture, and a body of knowledge that provide guidance for faculty who are designing and updating their programs. That guidance is intentionally flexible so that faculty can adopt and adapt it based on local programmatic needs. A reference curriculum is not intended to be used directly for program certification or accreditation.

requirement—Statement that identifies a product[9] or process operational, functional, or design characteristic or constraint, which is unambiguous, testable or measurable, and necessary for product or process acceptability (ISO/IEC 42010 2007).

service—A service can be defined as an activity required by one or more users who have agreed on the terms of outcomes and quality of service without details to how it is provided. A service is also, simply put, an act of help or assistance. In a more formal sense: Services are activities that cause a transformation of the state of an entity (people, product, business, and region or nation) by mutually agreed terms between the service provider and the customer (Chang 2010).

software engineering –the application of engineering methods to the development, operation, and maintenance of software.

system— (1) "A System is a set of related elements that form an integrated whole." A system exists in an **Environment** which contains related systems and conditions:

- **Closed Systems** have no relationships with the environment.
- **Open Systems** share **Inputs** and **Outputs** with its environment across the boundary.

System elements may be conceptual organizations of ideals in symbolic form or real objects, e.g. people, data, physical artifacts, etc.

- **An abstract** system contains only conceptual elements.

[9] "product" includes product, service, and enterprise.

- **A concrete** system contains at least two elements which are objects.

This simply idea is then further elaborated through a set of system principles and concepts. (von Bertalanffy 1968); (2) An entity which consists of three related sets: a set of elements; a set of internal interactions between the elements; and a set of external interactions between one or more elements and the external world; i.e. interactions that can be observed from outside the system. (Aslaksen 2004); (3) “An interacting combination of elements to accomplish a defined objective. These include hardware, software, firmware, people, information, techniques, facilities, services, and other support elements.” (INCOSE 2011).

system design and development (SDD) —One of the concentrations of the CorBoK. SDD is defined as performing system design and development related activities.

systems-centric—Term used to describe a SE master’s program which is focused on more general systems engineering concepts, knowledge, and skills, and is agnostic as to the field(s) in which these principles are applied. Described further in Fabrycky (2010). See also **domain-centric**.

systems engineering (SE)— (1) Any application of a combination of traditional engineering and holistic systems thinking, working with domain engineering, human sciences, management and commercial disciplines, to support the engineering of one or more systems of interest to come; (2) Interdisciplinary approach governing the total technical and managerial effort required to transform a set of customer needs, expectations, and constraints into a solution and to support that solution throughout its life; (3) 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: Operations, Performance, Test, Manufacturing, Cost & Schedule, Training & Support, Disposal. Systems engineering integrates all the disciplines and specialty groups into a team effort forming a structured development process that proceeds from concept to production to operation. Systems engineering considers both the business and the technical needs of all customers with the goal of providing a quality product that meets the user needs. (INCOSE 2010, 1)

system engineering management (SEM)—One of the concentrations of the CorBoK. SEM is defined as managing the resources and assets allocated to perform systems engineering activities..

systems thinking—(1) An epistemology which, when applied to human activity is based on four basic ideas: emergence, hierarchy, communication, and control as characteristics of systems (Checkland 1999); (2) A process of discovery and diagnosis – an inquiry into the governing processes underlying the problems and opportunities (Senge 1990); (3) A discipline for examining wholes, interrelationships, and patterns utilizing a specific set of tools and techniques (Senge 1990).

verification—(1) Confirmation, through the provision of objective evidence, that specified requirements have been fulfilled; (ISO/IEC 25000 2005, 4.64) (ISO/IEC 15288 2008, 4.38; ISO/IEEE 12207 2008, 1, 4.55); (2) Formal proof of program correctness; (ISO/IEC/IEEE 24765 2010, 1); (3) The process of evaluating a system or component to determine whether the products of a given development phase satisfy the conditions imposed at the start of that phase. (IEEE 1012 2004, 3.1.36) (4) Confirmation by examination and provision of objective evidence that the particular requirements for a specific intended use are fulfilled. (ISO/IEC 9126-1 2001) (5) The evaluation of whether or not a product, service, or system complies with a regulation, requirement, specification, or imposed condition. (PMI 2008) (6) Process of providing objective evidence that the software and its associated products comply with requirements (e.g., for correctness, completeness, consistency, and accuracy) for all life cycle activities during each life cycle process (acquisition, supply, development, operation, and maintenance), satisfy standards, practices, and conventions during life cycle processes, and successfully complete each life cycle activity and satisfy all the criteria for initiating succeeding life cycle activities. (IEEE 829 2008, 3.1.54) (7) Confirmation by examination and provision of objective evidence that specified requirements have been fulfilled. (ISO 8402 1994) (8) The process of ensuring that a system is built according to stakeholder requirements and design specifications (Buede 2009).

This page intentionally left blank.

References

Accreditation Board for Engineering and Technology (ABET). 2011. “ABET Glossary.” Accessed November 16. http://www.abet.org/glossary/.

Accreditation Board for Engineering and Technology (ABET) Engineering Accreditation Commission. 2010. “Criteria for accrediting engineering programs, 2011-2012.” Accessed November 16, 2011. http://www.abet.org/eac-current-criteria/.

Abran, A., J.W. Moore, P. Bourque, and R. Dupuis. 2004. “SWEBOK: Guide to the software engineering body of knowledge: 2004 version.” Los Alamitos, CA; Tokyo, Japan: IEEE Computer Society Press.

Academy of Program/Project & Engineering Leadership (NASA APPEL). 2009. *NASA's Systems Engineering Competencies*. Accessed on September 15, 2011. http://www.nasa.gov/pdf/303747main_Systems_Engineering_Competencies.pdf

ACM/IEEE-CS Joint Task Force on Computer Curricula. 2004. *Software engineering 2004, curriculum guidelines for undergraduate degree programs in software engineering*. Last modified August 25. http://sites.computer.org/ccse/.

Aslaksen, E.W. 2004. *A critical examination of the foundations of systems engineering tutorial.* Paper presented at 14th Annual International Council on Systems Engineering (INCOSE) International Symposium, Toulouse, France, June 20-24.

Bloom, B.S. 1994. “Reflections on the development and use of the taxonomy.” In *Bloom's taxonomy a forty-year retrospective*. eds. Lorin W. Anderson, Lauren A. Sosniak. Vol. 93, part 2, 1-8. Chicago, Illinois, USA: The University of Chicago Press.

Bloom, B.S., M.D. Engelhart, E.J. Furst, W.H. Hill, and David R. Krathwohl. 1956. *Taxonomy of educational objectives the classification of educational goals handbook I: Cognitive domai*n. London, England: Longman Group Ltd.

Brainerd, J.G. 1962. Editorial. *IRE Transactions on Educatio*n 5 (2): 55-6.

Buede, D. M. 2009. *The Engineering Design of Systems: Models and Methods*. 2nd ed. Hoboken, NJ, USA: John Wiley & Sons Inc.

Charette, R. N. 2009. “This car runs on code.” *IEEE Spectrum*. February 2009. Accessed November 16, 2011. http://spectrum.ieee.org/green-tech/advanced-cars/this-car-runs-on-code.

———. 2005. “Why software fails.” *IEEE Spectrum*. September 2005. Accessed November 17, 2011. http://spectrum.ieee.org/computing/software/why-software-fails

Chang, C. M. 2010. *Service Systems Management and Engineering: Creating Strategic Differentiation and Operational Excellence*. New York, NY, USA: John Wiley & Sons, Inc.

Chase, R., and N. Aquilano. 1995. *Production and operations managemen*t. Manufacturing and services. New York, NY, USA: Irwin/McGraw Hill.

Checkland, P. B. 1999. *Systems thinking, systems practic*e. Chichester, UK: John Wiley & Sons Ltd.

Defense Acquisition University (DAU). 2011. "Defense Acquisition University Acronyms Glossary and Terms' Thirteenth Edition, November 2009 as updated February 2011." https://dap.dau.mil/glossary/Pages/Default.aspx.

Engineering Council. 2011. "Chartered Engineer (CEng): About CEng." *Engineering Council* website. Accessed December 2011. http://www.engc.org.uk/professional-qualifications/chartered-engineer/about-chartered-engineer.

Fabrycky, W. J. 2010. *Systems engineering: It's emerging academic and professional attributes*. Paper presented at 2010 American Society of Engineering Education (ASEE) Annual Conference and Exposition, Lexington, KY, USA.

Ferris, T. and L. J. 2010. *Comparison of Systems Engineering Competency Frameworks*. Paper presented at 4th Asia-Pacific Conference on Systems Engineering (APCOSE), Keelung, Taiwan, October 4-6.

Fulks J. 2004 In Section 2 *Assessing Student Learning in Higher Education; Choosing the Right Assessment Tools.* Assessing Student Learning in Community Colleges. Last modified July 11, 2006. http://online.bakersfieldcollege.edu/courseassessment/Section_2_Background/Section2_2WhatAssessment.htm

Furst, E.J. 1994. Bloom's taxonomy: Philosophical and educational issues. In *Bloom's taxonomy a forty-year retrospective.*, eds. L.W. Anderson, L.A. Sosniak. Vol. 93, part 2, 28-40. Chicago, Illinois, USA: The University of Chicago Press.

Goda B., C. Reynolds. 2010. *Improving Outcome Assessment in Information Technology Program Accreditation*. Journal of Information Technology Education: Innovations in Practice, Volume 9, 2010. Santa Rosa, CA, USA: Informing Science Institute.

Gray, P. 1977. A system of training for general practice. *The Journal of the Royal College of General Practitioners: The British Journal of General Practice*. London, UK: The Journal of the Royal college of General Practitioners.

Gronlund, N. E. 2000. *How to write and use instructional objective*s. 6th ed. Upper Saddle River, NJ, USA: Prentice Hall (now Pearson Education).

Hamming, R. W. 1962. Computers and simulation in systems engineering. *IRE Transactions on Educatio*n 5 (2) (June): 76-8.

Howard, R.A. 1962. Mathematical programming in systems engineering. *IRE Transactions on Educatio*n 5

(2) (June): 78-81.

HR-XML Consortium. “HR-XML Competencies (Measurable Characteristics) Recommendation, 2004-08-02,” ed. Allen, C. Accessed October 30, 2011. http://ns.hr-xml.org/2_3/HR-XML-2_3/CPO/Competencies.html.

Huitt, W. 2006. The cognitive system. In *Educational psychology interactive*. Valdosta, GA, USA: Valdosta State University.

Institute of Electrical and Electronics Engineering (IEEE). 2004. *IEEE Standard for Software Verification and Validation.* IEEE 1012-2004. New York, New York, USA: Institute of Electrical and Electronics Engineers (IEEE) Standards Association.

Institute of Electrical and Electronics Engineering (IEEE). 2008. *IEEE Standard for Software and System Test Documentation*. IEEE 829-2008 : 3.1.6. New York, New York, USA: Institute of Electrical and Electronics Engineers (IEEE) Standards Association.

International Council on Systems Engineering (INCOSE). 2010. *INCOSE systems engineering handbook, version 3.2.* INCOSE-TP-2003-002-03.2. San Diego, CA, USA: International Council on Systems Engineering (INCOSE).

−−−. 2011. INCOSE *Systems Engineering Handbook, version 3.2.1*. International Council on Systems Engineering (INCOSE), INCOSE-TP-2003-002-03.2.1. San Diego, California, USA: International Council on Systems Engineering (INCOSE).

International Council on Systems Engineering (INCOSE) UK Systems Engineering Competencies Working Group. 2010. *Systems engineering competencies framework INCOSE-TP-2010-00*3. 3rd ed. San Diego, California, USA: International Council on Systems Engineering (INCOSE).

ISO. 2000. Quality management systems -- Fundamentals and vocabulary. ISO 9000.

ISO. 1994. Quality Management and Quality Assurance — Vocabulary Geneva, Switzerland: International Organization for Standardization (ISO). ISO 8402:1994.

ISO/IEC. 29881. 2008. *Information Technology — Software and Systems Engineering — FiSMA 1.1 Functional Size Measurement Method*. ISO/IEC. : A.4. Geneva, Switzerland: International Organization for Standardization (ISO)/International Electrotechnical Commission (IEC).

———. 2007. *Systems and software engineering -- recommended practice for architectural description of software-intensive systems.* ISO/IEC 42010:2007. Geneva, Switzerland: International Organization for Standards (ISO)/International Electronical Commission (IEC).

———. 2005. *Software Engineering — Software Product Quality Requirements and Evaluation (SQuaRE) -- Guide to SQuaRE.* ISO/IEC 25000:2005. Geneva, Switzerland: International Organization for

Standardization (ISO)/International Electrotechnical Commission (IEC).

———. 2001. *Software engineering — Product quality — Part 1.* ISO/IEC 9126-1:2001. Geneva, Switzerland: International Organization for Standardization (ISO)/International Electrotechnical Commission (IEC).

———. 1998. *Information Technology — System and Software Integrity Levels.* ISO/IEC. 15026: 1998: 3.1. Geneva, Switzerland: International Organization for Standardization (ISO)/International Electrotechnical Commission (IEC).

ISO/IEC/IEEE. 2010. "Systems and Software Engineering — Vocabulary (SEVocab)." ISO/IEC/IEEE 24765:2010. Geneva, Switzerland: International Organization for Standardization (ISO)/International Electrotechnical Commission (IEC)/Institute of Electrical and Electronic Engineers (IEEE) 2009. http://pascal.computer.org/sev_display/index.action.

ISO/IEC/IEEE. 2008. *Systems and software engineering - system life cycle processes.* ISO/IEC 15288:2008 (E). Geneva, Switzerland: International Organization for Standardization (ISO)/International Electronical Commission (IEC).

Jain, Rashmi, A. Squires, D. Verma, and A. Chandrasekaran. 2007. A reference curriculum for a graduate program in systems engineering. *Insigh*t 10 (3) (July): 9-11.

Jain, Rashmi, and D. Verma. 2007. *A report on curriculum content for a graduate program in systems engineering: A proposed framework INCOSE-PP-2007-001-0*1International Council on Systems Engineering (INCOSE).

Karlin, John E. 1962. Human factors engineering and research in telephone systems engineering. *IRE Transactions on Educatio*n 5 (2) (June): 71-5.

Kelly, Mervin J. 1950. The bell telephone laboratories - an example of an institute of creative technology. *Proceedings of the Royal Society of London*. Series B 137 (889) (November): 419-33.

Kirkpatrick, D. L. 1994 *Evaluating training programs: the four levels*. San Francisco, CA, USA: Berrett-Koehler.

Krathwohl, D. R. 2002. A revision of bloom's taxonomy: An overview. *Theory into Practic*e 41 (4): 212-8.

Krathwohl, D.R., B.S. Bloom, and B.B. Masia. 1964. *Taxonomy of educational objectives the classification of educational goals handbook II: Affective domai*n. New York, USA: David McKay Company, Inc.

Kreitzer, Amelia E. and G.F. Madaus. 1994. Empirical investigations of the hierarchical structure of the taxonomy. In *Bloom's taxonomy a forty-year retrospective*. eds. Lorin W. Anderson, Lauren A. Sosniak. Vol. 93, part 2, 64-81. Chicago, Illinois, USA: The University of Chicago Press.

Lewy Arieh and Z. Bathory. 1994. The taxonomy of educational objectives in continental Europe, the

Mediterranean, and the Middle East. In *Bloom's taxonomy a forty-year retrospective*, eds. Lorin W. Anderson and Lauren A. Sosniak. Vol. 93, part 2, pp. 146-163. Chicago, Illinois, USA: The University of Chicago Press.

Linvill, William K. 1962. Models and model construction. *IRE Transactions on Educatio*n 5 (2) (June): 64-7.

Matchett, Gerald J. 1962. Relation of engineering economy to systems engineering. *IRE Transactions on Educatio*n 5 (2) (June): 85-6.

Merrison Committee. 1975. *Report of the committee of inquiry into the regulations of the medical profession (report of the royal commission on the national health service).* London, England: Her Majesty's Stationary Office (HMSO), Cmnd. 6018.

Mooz, H., Forsberg, K., Cotterman, H. 2003. *Communicating Project Management*. Hoboken, NJ, USA: John Wiley and Sons.

Nara, H. R. 1962. Systems engineering and research at case institute of technology. *IRE Transactions on Educatio*n 5 (2) (June): 105-8.

PMI. 2008. *A guide to the project management body of knowledge (PMBOK® guide*). 4th ed. Newtown Square, PA, USA: Project Management Institute (PMI).

Professional Engineer, The. 2011. "Requirements." *The Professional Engineer.com*. Accessed December 2011. http://theprofessionalengineer.com/requirements/.

Prus J., R. Johnson . 1994. A Critical Review of Student Assessment Options. In *Assessment & Testing: Myths and Realities. New Directions for Community Colleges, Number 88. e*ds. by Trudy H. Bers and Mary L. Mittler. pp. 69-83. San Francisco, CA, USA: Jossey-Bass, Inc.

Pyster, A., ed. 2009. *Graduate software engineering 2009 (GSwE2009) - curriculum guidelines for graduate degree programs in software engineering, version 1.*0. Integrated Software & Systems Engineering Curriculum Project. Hoboken, NJ, USA: Stevens Institute of Technology.

Pyster, A., D. Olwell, A. Squires, N. Hutchison, S. Enck and J. Anthony, eds. Released for limited review 2011. *A guide to the systems engineering body of knowledge (SEBoK) version 0.50*. Body of Knowledge and Curriculum to Advance Systems Engineering (BKCASE). Hoboken, NJ, USA: Stevens Institute of Technology. Available at sebokwiki.org (accessed November 29, 2011)

Reitman, W.R. 1962. Systems engineering in interdisciplinary curricula - the program at Carnegie Institute of Technology. *IRE Transactions on Educatio*n 5 (2) (June): 119-21.

Rohwer, W.D., and K. Sloane. 1994. Psychological perspectives. In *Bloom's taxonomy a forty-year retrospective*., eds. L.W. Anderson, L.A. Sosniak. Vol. 93, part 2, 41-63. Chicago, Illinois, USA: The

University of Chicago Press.

Schmidt, S.P. 1975. An evaluation of teaching method in theological education. In *Voice of the church in Asia*., ed. Bong Rin Ro, pp. 36-49. Singapore: Asia Theological Association.

Senge, P. M. 1990. The fifth discipline: The Art & Practice of the Learning Organization. New York, NY, USA: Doubleday Business.

Showers, R.M. 1962. Systems engineering at the Moore school of electrical engineering. *IRE Transactions on Educatio*n 5 (2) (June): 113-5.

Sosniak L. A. The taxonomy, curriculum, and their relations. *Bloom's taxonomy a forty-year retrospective*. vol. 93, part 2, L. W. Anderson and L. A. Sosniak, Eds., ed Chicago, Illinois, USA: The University of Chicago Press, 1994, pp. 103-125.

Squires, A, D. Olwell, T. Ferris, N. Hutchison, A. Pyster, S. Enck, and D. Gelosh. 2011. *AC 2011-958: Developing Systems Engineering Graduate Programs Aligned to the Body of Knowledge and Curriculum to Advance Systems Engineering (BKCASE™) Guidelines.* Paper presented at the American Society for Engineering Education (ASEE), Vancouver, B.C., Canada, June 26-29.

Squires, A, W. Larson, and B. Sauser. 2010. Mapping space-based systems engineering curriculum to government-industry vetted competencies for improved organizational performance. *Systems Engineerin*g 13 (3): 246-260. Hoboken, NJ, USA: John Wiley & Sons.

Squires, A. and R. Cloutier. 2010. Evolving the INCOSE reference curriculum for a graduate program in systems engineering. *Systems Engineerin*g 13 (4): 381-388. Hoboken, NJ, USA: John Wiley & Sons.

Squires, A and W. Larson. 2009. Improving Systems Engineering Curriculum using a Competency-Based Assessment Approach, International Journal of Intelligent Defence Support Systems, Vol. 2, No.3 pp. 184 – 201. Olney, Buckinghamshire, UK: Inderscience Publishers.

Truxall, J.G. 1962. Feedback and systems engineering. *IRE Transactions on Educatio*n 5 (2) (June): 82-5.

Tufts, B. 2002. Model Curriculum for Master of System Engineering Degree. *INCOSE.* San Diego, CA, USA: International Council on Systems Engineering (INCOSE).

UCF. 2008. *Program assessment handbook - guidelines for planning and implementing*. Orlando, FL, USA: University of Central Florida (UCF).

United States Department of Education (USDOE). 2011. “ED.gov.” Last modified November 17. http://www2.ed.gov/admins/finaid/accred/accreditation.html#Overview.

VanLeer, M. 2007. *Systems Engineering Competency Development*. Paper presented at the 5th Annual Conference on Systems Engineering Research, Stevens Institute of Technology, Hoboken, NJ, USA March 14-16.

von Bertalanffy, L. 1968. *General System Theory: Foundations, Development, Applications.* Revised ed. New York, NY, USA: George Braziller, Inc.

Wiggins, G. 1993. *Assessing student performance: Exploring the purpose and limits of testing*. Hoboken, NJ, USA: John Wiley & Sons (Jossey Bass).

Wisconsin. 2009. "The University of Wisconsin Madison: Outcomes assessment. " Last modified November 25. http://www.provost.wisc.edu/assessment/manual/manual2.html

Wymore, A. W. 1962. The systems engineering department, college of engineering, University of Arizona. *IRE Transactions on Education* 5 (2) (June): 108-12.

Yarbrough, D. B., L. M. Shulha, R. K. Hopson, and F. A. Caruthers. Scheduled. 2011. *The program evaluation standards: A guide for evaluators and evaluation users*. 3rd ed. Thousand Oaks, CA, USA: Sage - Joint Committee on Standards for Educational Evaluation.

This page intentionally left blank.

Appendix A: Context and Guidance for the Construction and Maintenance of GRCSE

A.1 Background

The *Graduate Reference Curriculum for SE* (GRCSE) author team is comprised of almost 70 authors from around the world. GRCSE is intended to be globally applicable and represent the current views of the systems engineering (SE) community regarding professional master's education in SE. In order for a disparate group to accomplish this task, it was critical to develop a set of principles to guide the development of GRCSE. This appendix describes the foundational guidance adopted in developing GRCSE. In Chapter 1, there are four primary assumptions regarding the development of GRCSE. These assumptions are further fleshed out in this appendix. Chapter 1 also provides a discussion of GRCSE guidance, specifically focusing on broad contextual issues. Additional depth on each of these principles is provided here.

This guidance derives from the principles stated in *Graduate Software Engineering 2009* (GSwE2009*)* (Pyster (ed) 2009), which in turn was influenced by *Software Engineering 2004* (SE2004) (ACM/IEEE-CS Joint Task Force on Computer Curricula 2004). Differences appear where this guidance distinguishes the higher expectations of graduate education from those of undergraduate education and explicitly recognizes the ties between SE and other disciplines. In many cases, this guidance could apply to curricula for virtually all engineering disciplines, not just SE. This is not unexpected, given the commonalities between SE and other engineering disciplines.

A.2 Assumptions

This section presents four assumptions about SE that have shaped the development of GRCSE.

1. *SE is a distinct discipline with a rich body of knowledge, practice, and theory whose essential guide is the Body of Knowledge and Curriculum to Advance Systems Engineering's (BKCASE) Guide to the Systems Engineering Body of Knowledge (SEBoK).*

   Since 1950, when the term "SE" was first published (Kelly 1950), the growth of the discipline is seen in the numerous journals, conferences, professional societies, undergraduate and graduate programs, professional certifications, standards, and large cadre of faculty and practitioners that claim to address SE. The SEBoK represents a guide to the community of this rich body of knowledge. Direct linkages to the SEBoK are established throughout GRCSE, specifically in Chapter 2 (outcomes) and Chapter 6 (the *Core Body of Knowledge* (CorBoK)).

2. *SE curricula must recognize the interconnections between the many disciplines from which SE draws its foundations, including software engineering, human factors, project management, and the various engineering specialties.*

   Graduate study in SE typically draws from many areas in engineering and computing for its theoretical and conceptual foundations. It also draws from other fields, including statistics, calculus, other mathematical specialties; technical management, professional communication, leadership, and project management; economics; systems science; the social sciences; and one or more application domains. Chapter 6 presents the CorBoK, which outlines the topics recommended by GRCSE that stem from other relevant disciplines as well as the connections between these disciplines and SE.

3. *There is no generally recognized graduate reference curriculum for SE, and there is wide variation in existing programs.*

   The GRCSE team conducted a survey of more than 30 SE master's programs worldwide, and confirmed that often there is a low level of commonality between programs (please see Appendix B: Summary of Graduate Systems-Centric SE Programs in 2010 for additional information). This lack of commonality means that systems engineers from different programs may have widely varying skills, which makes it difficult for employers to understand what capabilities a potential employee has when they possess a master's degree in SE. By improving commonality between programs at some level, it is believed that the capabilities of the SE workforce as a whole may be improved. This assumption is further elaborated in Chapter 1.

   It is important to note, however, that while some commonality is desired, an appropriate level of diversity is also critical to the development and evolution of a successful SE workforce. This balance may be seen, for example, in the GRCSE architecture and the CorBoK, which recommend that approximately half a master's curriculum should be unique to a specific program.

4. *All SE students must learn to integrate theory and practice.*

   For graduates to make sound and appropriate contributions in the workplace, they must be adept at synthesizing the theoretical and conceptual with the practical aspects of SE. This assumption is seen in the recommended student outcomes (Chapter 3), which state that students should achieve a Bloom's synthesis level in at least one area during their master's education.

A.3 Principles

These principles are viewed as requirements for the GRCSE team's efforts, because failure to follow these principles may result in a reference curriculum that cannot meet the stated purpose of GRCSE (for additional information on the purpose, please see Chapter 1). Because these requirements apply to the development of each iteration of GRCSE, they are written in future tense. A description of how GRCSE satisfies a principle and the rationale for the chosen approach is provided within the discussion of each principle.

The principles have been kept general to provide flexibility because GRCSE is a reference curriculum that must remain independent of local conditions even as it supports tailoring to accommodate local needs. Local factors which can affect the design of any particular program include

- institutional policies concerning program design;
- regulatory issues related to education program design;
- norms of preparation for entrants to the program; and

- regional SE needs.

The reader should keep this context in mind while reviewing the principles. (For more information on tailoring for local factors, please see Chapter 7.)

The principles are grouped based on subject matter into three categories: scope, program architecture, and context. The principles are numbered within these categories for reference, but their ordering does *not* imply importance or precedence.

A.3.1 Scope

1. *GRCSE will provide a set of adaptable recommendations for developing and improving curricula for SE education at the master's degree level. It is* ***not*** *developed as a basis for accreditation, certification, or licensing.*

   It is important to note that accreditation, certification, and licensure are not within the scope of GRCSE. However, the GRSCE team believes that if GRCSE is successfully adopted by the community, it will be used as an input to organizations which develop and maintain standards for accreditation, certification, and licensure.

2. *The master's degree described by GRCSE will be a professional degree concentrated on enhancing the skills and knowledge of practicing systems engineers.*

   The vast majority of students who earn a master's degree in SE do not become researchers. Most SE graduate students are, or aspire to become, practicing professionals. These students seek to broaden and improve their skills and on-the-job opportunities. GRCSE will target the education of these practicing professionals. This has directly influenced the expected outcomes for GRCSE, which are clearly focused on the skills required for practicing professionals. With modification, however, GRCSE may serve as a suitable foundation for those with research interest or the desire to proceed to a doctoral degree in SE.

3. *GRCSE will be broadly based and international in scope.*

   GRCSE is intended to support the needs of industry and government by helping universities to equip systems engineers with contemporary theory and practice and to develop their ability to address the future challenges of systems development. As a reference curriculum, GRCSE provides a set of recommendations for use in constructing curricula for a master's degree in SE. It is *not* a standard and it is expected that GRCSE will be tailored to address the particular needs and context of each program. It should be noted, however, that while GRCSE is intended to be tailored, it would not be appropriate for a program to claim alignment with GRCSE if it has been tailored so dramatically that it does not adequately satisfy the spirit of the GRCSE recommendations. GRCSE reflects the collective wisdom of a broad community of authors and reviewers about graduate SE education and is intended to help the community gain some consistency in graduate SE education.

   Curricular requirements and educational expectations differ between countries, but the problems facing SE have common themes. Where appropriate, every effort has been made to ensure that the curriculum recommendations are sensitive to national and cultural differences so that they are internationally applicable. The diversity of the GRCSE authors has supported the development of an internationally relevant reference curriculum.

A.3.2 Program Architecture

1. *A master's program that satisfies the GRCSE recommendations should require approximately the same amount of study[10] as programs do now.*

   The intent is for GRCSE to fit within the common scale of currents graduate programs; a program should not have to be significantly lengthened in order to follow the GRCSE recommendations. For simplification, it is assumed that a master's degree requires 450- 500 contact hours. This limitation forms an underlying assumption for the GRCSE outcomes and specifically impacts the structure of the CorBoK recommendations. It also indirectly impacts the entrance expectations. More program time could be required to achieve more in-depth outcomes; since programs should be the same length, student capabilities upon entry become more critical. This has specifically influenced the entrance expectation for two years of professional experience (for additional information on entrance expectations, please see Chapter 4.)

2. *GRCSE will identify the knowledge and experience expected for students to enter a master's program in SE.*

   Because student ability on entrance is an important input to a graduate program, it is important for GRCSE to set some expectations about student capabilities. Undergraduate programs and industry experience vary greatly in the nature of the SE background they provide to program entrants. To help institutions build programs that address the needs of the broad SE community, GRCSE recommends a minimum prerequisite knowledge and experience. This minimum prerequisite knowledge and experience is determined by the level of proficiency to be achieved by the time of entry to the master's program. This is addressed in Chapter 4 of GRCSE.

3. *GRCSE will be based on a flexible curriculum architecture and the SEBoK.*

   GRCSE's basis on the SEBoK aligns with the first assumption (above). Flexible curriculum architecture is important if GRCSE is to be applicable across many different educational systems in many different countries. Chapter 5 describes the recommended program architecture for GRCSE.

[10] Different universities and education systems have different methods to express quantities of study. For example, universities use terminology such as "units", "credits", "hours", or simply require a certain list of courses to be completed. Even universities which use the same words may indicate a different amount of effort for a quantity of "one" of something. Therefore it is impractical to attempt to describe quantities of study in terms of such terminology. For this reason, the GRCSE uses the concept of effort associated with "full-time equivalent" study, the amount of effort a student would be expected to expend if studying the program on a full-time basis. From this notional count of time, it is possible take proportions which meaningfully indicate amounts of effort devoted to particular parts of a program by reference to a proportion of a whole program.

4. *GRCSE will support individual program and student flexibility by limiting the core foundation and concentrations required for all students to 50% of the instructional time in a master's program.*

   A curriculum requires flexibility to tailor the breadth and depth of the requisite knowledge and to develop the desired skills in students. Universities will individually decide how to package the teaching and learning experiences into courses. The finite time available for a master's degree necessarily constrains the expectations about the amount of learning expected in the GRCSE recommendations. GRCSE also recognizes that different education systems make different constraints on the amount of study expected in a master's degree program (see program architecture guiding principle 1, above). The GRCSE approach defines a core that can be achieved in about half the normal duration of master's degrees, reserving time for university specific content. This provides program flexibility for defining its ways to achieve its outcomes and objectives.

   GRCSE expects that curriculum developers will design their programs cognizant of the actual preparation of their typical entrants and develop teaching and learning experiences that enable the students admitted to the program to develop the intended knowledge and skills through the program.

A.3.3 Context

1. *GRCSE will provide for ongoing review and revision of its reference curriculum to reflect the rapid evolution and professional nature of SE.*

   Universities, industry, and governments, in cooperation with professional societies in SE, must establish an ongoing maintenance process that allows individual components of the curriculum recommendations to be updated on a recurring basis.

   The BKCASE team expects the International Council on Systems Engineering (INCOSE) and the IEEE Computer Society to take stewardship of GRCSE after the release of Version 1.0. Part of this stewardship includes a commitment to continuing to collect community feedback on GRCSE and updating the reference curriculum periodically to better reflect current practice and needs.

2. *GRCSE will include discussions of strategies and tactics for implementation along with high-level recommendations.*

   Although it is important for GRCSE to articulate a broad vision of SE education, the success of any real university curriculum depends heavily on the implementation details. Version 0.25 was not mature enough to include these considerations. However, the current Version 0.5 includes the draft guidelines for implementing GRCSE while developing a new program or within an existing program. This material is presented in Chapter 7. However, it is anticipated that this material will undergo heavy updating between Version 0.5 and Version 1.0.

3. *GRCSE will include non-technical aspects of professional practice as an integral component of the graduate curriculum.*

The professional practice of SE encompasses a wide range of non-technical issues and activities, including general problem solving; management; ethical and legal concerns; the communication skills of reading, writing, listening and speaking; working as part of a team; incorporating an understanding of psychology; ethnic culture and religious diversity; ethical behavior; and recognizing the need for other expertise in a rapidly changing discipline. These are professional attributes that enable the systems engineer to ensure that suitable solutions to needs are delivered. These issues and activities are addressed in the objectives (Chapter 2), outcomes (Chapter 3), and the CorBoK (Chapter 6).

4. *GRCSE will be sensitive to changes in technologies, practices, and applications in SE, to new developments in pedagogy, and to the importance of lifelong learning.*

   The principles underlying SE change relatively slowly, but the technology through which SE is practiced changes rapidly. Educational institutions must adopt explicit strategies for responding to changing technology without simply “training” in the latest technology. A key to this is organizing the curriculum around enduring principles and planning to change the supporting technologies regularly. The CorBoK (Chapter 6) provides a framework to assist universities in maintaining clarity concerning the enduring principles that should be enabled and supported through programs. This perspective is particularly important in SE, where the systems engineer’s role is to provide leadership in the development and integration of innovative technologies into complex systems. In addition, this principle is specifically addressed in the expected outcomes (Chapter 3).

5. *GRCSE will identify the fundamental skills, attributes, and knowledge that all graduates of a SE master’s degree program must possess.*

   GRCSE defines outcomes that all graduates should achieve. They range from the highly technical to the “soft skills” involving communication and ethics and the qualities associated with leadership. Additionally, GRCSE defines a specific foundation that every student should master by graduation. That knowledge contributes to the achievement of the 13 outcomes. However, only students who go well beyond the content of the foundation can achieve all the outcomes. Consequently, the university specific portion of each program will be an important contributor to the achievement of the outcomes. The knowledge critical for graduates of SE master’s programs is defined within the CorBoK (Chapter 6).

6. *GRCSE will be informed by the competencies identified as necessary or desirable by employers of systems engineers and by professional associations of systems engineers.*

   The GRCSE team has worked with several professional societies in the development of the recommendations contained herein, including INCOSE, the IEEE Systems Council and Computer Society, the Association for Computing Machinery (ACM), and the National Defense Industrial Association’s (NDIA’s) SE Division. Representatives from these professional societies provide insight into the needs of potential SE employers. In addition, participation by representatives of government and industry on the BKCASE author team has helped to ensure that GRCSE addresses the needs of both these employer bases. Finally, reviewers are asked to provide further response and insight into how well GRCSE 0.5 responds to these needs, so that GRCSE 1.0 will address them more fully.

7. *GRCSE will be informed by the perspective that both education and competencies concern the whole person of the student, including the cognitive and affective domains.*[11]

   Appendix C: Bloom's Taxonomy of Educational Outcomes provides explanations of the cognitive and affective domains of Bloom's taxonomy. The CorBoK utilizes the cognitive domain taxonomy to provide guidance on what students should be able to do with the knowledge associated with specific topics. The affective domain will be more cleanly woven into GRCSE for Version 1.0.

8. *GRCSE will maintain a clear distinction between itself and the SEBoK product of the BKCASE™ project.*

   CorBoK relates to a subset of the knowledge discussed and organized in the SEBoK. The GRCSE description of content for graduate programs is more narrowly focused than the SEBoK because GRCSE includes the foundational knowledge required for a person to commence practice whereas the SEBoK is a guide to the whole body of knowledge of SE. GRCSE expects considerable sophistication in student reasoning about the use of SE principles and also expects students to demonstrate their accumulated skills and knowledge in a significant capstone experience (project, practicum, or thesis). GRCSE uses Bloom's taxonomy to describe the minimum levels of achievement expected in the various topic areas. (See Appendix C: Bloom's Taxonomy of Educational Outcomes for more information.)

[11] The idea of the "Cognitive Domain" and the "Affective Domain" are drawn from Bloom's taxonomy of educational outcomes (Bloom et al. 1956; Krathwohl, Bloom, and Masia 1964). Bloom's taxonomy is used in the GRCSE as outlined and explained in Appendix C: Bloom's Taxonomy of Educational Outcomes.

This page intentionally left blank.

Appendix B: Summary of Graduate Systems-Centric SE Programs in 2010

B.1 Methodology

As *the Graduate Reference Curriculum for Systems Engineering* (GRCSE) authors began considering the appropriate recommendations for outcomes, objectives, and entrance expectations, and the appropriate materials to include within the *Core Body of Knowledge* (CorBoK), they realized that it was important to understand the current landscape of systems engineering (SE) master's programs. With this baseline, they could better understand what recommendations were appropriate for GRCSE. For example, recommendations less stringent than what is seen in many programs today would not benefit the community, while recommendations that are far more rigorous than anything seen today might be considered unattainable. Understanding the baseline of current programs, then, was a critical step in moving forward with the GRCSE project.

In the spring and summer of 2010, an international survey was conducted of Master's of SE (MSE) programs. Survey forms were sent to 59 schools, which offered a combined total of 63 MSE programs. The initial list of candidate US schools and graduate programs was obtained from work performed by Fabrycky (2010). International members of the GRCSE team augmented this list with an additional 26 academic institutions outside the US. Survey responses were received for 33 programs from the following countries: Australia, Brazil, Canada, Cyprus, Finland, France, Japan, Singapore, Taiwan, and the United States (US). The range of schools that responded to the survey included traditional universities and specific-purpose institutions (such as the postgraduate institutions and alliances).

A survey instrument, similar to the one shown in Table 18 was developed and administered to collect and organize relevant data concerning the attributes of the program. Although attempts were made to standardize the way in which the data was provided, there are still some differences in the level of detail provided and the interpretation of the instructions by the academic program personnel. This required the team to make adjustments in the way they analyzed the data. The analysis can be found in Section B.2, below.

Table 18. Survey Data Collection Instrument (table developed for BKCASE).

Program Contact Information	
Name of Respondent	
University	
Name of Program	
Program Attributes	
Delivery Mode	Face to Face Online Hybrid Synchronous/asynchronous Multi-modal

Typical Course Module	Class meets for 12 weeks – once a week for 3 hours Individual tutorial by appointment Independent study Group Project
Program Length	Number of courses/units/credit hours required Number of full-time equivalent years typically required to complete the program
Program Focus	General System Focus Domain-Centric (e.g., biological, manufacturing, transportation)
Project/Thesis requirement	Capstone project course Thesis Internship Other Special Features
Requirements for Admission into the Program	
Undergraduate Degree	Acceptable undergraduate degrees
Undergraduate Performance	Acceptable undergraduate GPA
Exam Performance	Admission Exam GRE Score
Experience	Years practicing in the field
Language Proficiency	Test of English as a Foreign Language (TOEFL) International English Language Testing System (IELTS) Other exam
Program Outcomes	
Program Outcomes	List the expected outcomes of students upon graduation from the program.
Program Courses	List of typical courses a student might complete, with a short description of each course. Designate the required and elective courses. Provide some indication of the course magnitude (credit hours, contact hours, effort hours.)

B.1 Findings

It is important to note that, based on the survey responses, there is a high level of variation found in these programs. The spectrum of programs investigated led to a number of findings about the student acceptance criteria, completion criteria, general focus, and core courses of the programs. The initial findings of the survey are presented in Table 19. Following the table, there is a list of additional findings;

these findings highlight some of the discrepancies between programs as well as some of the commonalities.

Table 19. Survey Results (table developed for BKCASE).

Program Attributes	
Delivery Mode	
Face to Face	94%
Online	41%
Hybrid	25%
Multi-modal	50%
Average Course/Module Session	16 weeks
Average Program Length	2 years
Project Thesis or Capstone Project required	94%
Program Focus	
General Systems Focus; SE; Systems Architecture; Systems Design; Systems Management	66%
Domain Specific	34%
Requirements for Admission	
Accepted Undergraduate Degrees in one or more of the following areas	
Engineering	88%
Science/Physics/Mathematics	81%
Other	30%
Undergraduate Performance	
Average GPA 3.0	60%
No Requirement	28%
No Response	12%
Exam Performance Required? (e.g., the GRE)	
Yes	28%
No	50%
No response	22%
Language Proficiency TOEFL or IELTS	
Yes	69%
No	19%
Other	6%
No response	6%

Related Work Experience	
Average 3 years	40%
No Requirement	44%
Company Sponsored	16%
Top 10 Courses* Identified within the Program	
Introduction to SE	56%
Modeling and Simulation	53%
Project Management	38%
Systems Architecture and Design	38%
Systems Integration	34%
Systems Analysis	25%
Systems Management	25%
Systems Requirements Analysis	25%
Risk and Decision Analysis	25%
Probability and Statistical Analysis	19%

*These courses were identified using the course title provided by each university. Though the course titles may have differed, the GRCSE team was able to group the courses into several categories. These categories are represented here.

Interesting findings not represented in the above table include differences between the US-based and non-US based programs, as well as other matters described below.

- The findings show that 71% of non-US programs and 33% of US-based schools used only face-to-face interaction in the delivery of their courses. In addition, 67% percent of US-based institutions are utilizing technology in the delivery of their programs. These variances in delivery mode require the GRCSE guidelines to be broad enough to accommodate a variety of teaching and learning styles and various pedagogies.
- Almost all programs required the students to complete a thesis or capstone project. Of the respondents, only six programs specifically stated that the project must be completed in teams.
- Most US-based schools surveyed required an undergraduate grade point average (GPA) of 3.0 or better. In contrast, 79% of the non-US based schools had no requirement for an undergraduate GPA or equivalent. Because of this discrepancy in requirements, GRCSE contains an entrance expectation related to the type of undergraduate degree for students, but not for a specific GPA.
- Graduate entrance exams, such as the Graduate Records Exam (GRE) in the US, were not generally required outside of the US. Only one non-US university responded with a possible requirement for an entrance examination.

- Language proficiency tests, such as the Test of English as a Foreign Language (TOEFL) and the International English Language Testing System (IELTS), were required primarily by US-based schools that accept international students. In addition, the TOEFL and IELTS were requirements by non-US based schools specifically identifying their course delivery to be in English.
- Of the programs surveyed, 16% reported support by a company sponsor, with the majority of those being aligned with a national defense program.
- Program goals or outcomes statements, in a majority of instances, focused on preparing the student to develop, create, or innovate the next generation of systems or services.
- A majority of the programs identified “Introduction to SE” and “Modeling and Simulation,” or an equivalent, as a required course.

B.3 Conclusions

The initial survey and analysis work produced a reasonable profile of current master’s programs. Section B.2 presents a summary view of the SE master's programs that returned the survey. This view provided the author team with an idea of the current state of MSE graduate education, and hence, context for all its analysis, deliberations, and decisions. The clearly evident diversity that has also been reported in other research (Squires et al. 2010) has helped to motivate and inform the GRCSE effort.

This page intentionally left blank.

Appendix C: Bloom's Taxonomy of Educational Outcomes

This appendix provides a general discussion of Bloom's taxonomy of educational outcomes to provide a foundation for the use of Bloom's taxonomy in other parts of the *Graduate Reference curriculum for Systems Engineering* (GRCSE). The major place where Bloom's taxonomy and levels of attainment, commonly referred to as Bloom's levels, are used in GRCSE is in the description of the *Core Body of Knowledge* (CorBoK) in Chapter 6. This appendix provides the foundation for understanding and interpreting the intended levels of achievement indicated in the CorBoK.

C.1 Introduction to Bloom's Taxonomy

Bloom's taxonomy is a classification system, devised in 1956 by a group of educators led by Benjamin Bloom, for the purpose of providing a scientific foundation for curriculum design (Bloom et al. 1956; Krathwohl, Bloom, and Masia 1964) and was originally developed to categorize test items for exchange between university faculty members (Krathwohl 2002). The intent was to create a bank of test items that were categorized based on the highest level of what was being tested for each educational objective: knowledge, comprehension, application, analysis, synthesis, and evaluation. These educational objectives became the six levels of Bloom's taxonomy.

Bloom's taxonomy has stood the test of time. Over 50 years later, educators continue to use the taxonomy to set the level of attainment of educational or learning outcomes required for students engaged in an education unit, course, or program. The taxonomy has become commonly used in many countries, at multiple levels of education, and in many fields of teaching (Bloom 1994, 1-8). Because the taxonomy is known and understood globally to a greater extent than any of its competitors (Lewy and Bathory 1994), it is appropriate to use Bloom's taxonomy in an international document such as GRCSE.

Bloom's taxonomy divides educational outcomes into three domains: cognitive, affective, and psychomotor. The original 1956 work only developed taxonomies for the cognitive and affective domains. In GRCSE, the major focus is on the *cognitive domain,* which is concerned with knowledge and how it is learned (Huitt 2006). GRCSE also has a minor focus on the *affective domain,* which is concerned with feelings, appreciation, and valuation of the content that is learned (Schmidt 1975, 36-49). Conventional education systems tend to stress outcomes in the cognitive domain. In particular, conventional educators focus on lower-level cognitive outcomes and often do not explicitly develop higher-level cognitive outcomes. Additionally, these educators give little attention to affective domain engineering outcomes. This perspective is reflected in the CorBoK as presented in Chapter 6.

However, some types of education, for example military and theological, make the affective domain an explicit focus of their outcomes (see Chapter 3). GRCSE includes consideration of the affective domain in its guiding principles (see A.3 Principles, Context Guiding Principle #7), and explicitly discusses the relationships of the affective domain to systems education in this appendix. The role of the systems engineer is to lead the development of systems. This role includes working with engineered systems, deliberately taking a systems perspective, and negotiating solutions with multiple, diverse stakeholders. These requirements on the systems engineer make their proficiency in the attributes of the affective domain critical to their success. The necessity to deliberately take a systems perspective on the engineering activity demands the development of an affective perspective so that the systems engineer naturally chooses to address needs in the whole system context. The negotiation role involves making judgments about what is good or desirable, and demands that systems engineers personally value the systems perspectives required to perform their work.

Bloom's taxonomy is hierarchical: learning at a higher level is dependent on attaining prerequisite knowledge and skills at the lower levels. This characteristic of Bloom's taxonomy reflects its development in the framework of behaviorist psychology, which was dominant in the 1950s. The behaviorist links of Bloom's taxonomy reflect a tacit philosophy of education based on the development of desired behaviors in which the ability to perform an action is accepted as evidence of learning (Furst 1994, 28-40; Rohwer and Sloane 1994, 41-63). While Bloom's taxonomy has been challenged more recently because of the decline in the position of behaviorist psychology, it has remained popular among educators in pragmatic fields (such as systems engineering (SE)) where the goal of education is to produce graduates who can use a theoretical foundation to perform work (Kreitzer and Madaus 1994, 64-81).

C.2 Bloom's Taxonomy Levels in the Cognitive Domain

Table 20 provides a description of Bloom's taxonomy for the cognitive domain. The levels of the taxonomy are commonly referred to as Bloom's levels, and in a general context, the term "Bloom's level's" typically refers to the cognitive (as the default) rather than the affective domain. There is some debate about the ordering of the two highest cognitive domain levels, "Synthesis" and "Evaluation," specifically whether their order should be reversed (Krathwohl 2002) or whether they should be combined into a single level. This is an area for further research that the GRCSE project does not attempt to address. That is, with regards to reversing "Synthesis" and "Evaluation," GRCSE maintains the traditional Bloom's taxonomy order rather than the revised Bloom's taxonomy order (Krathwohl 2002). The challenge to the order of the levels depends on whether the order of the levels in the cognitive domain reflects a taxonomy of cognitive levels of attainment, but not necessarily of educational outcomes. This argument is not addressed in GRCSE because the purpose of GRCSE is to provide a reference curriculum for education specifically designed to lead to the capability to take action in a workplace setting, and the levels reflect levels of attainment that are relevant to the workplace in the logical order in which the capabilities are needed (Furst 1994).

GRCSE also maintains the traditional view that "Synthesis" and "Evaluation" are different kinds of attainment. In relation to engineering work, synthesis is concerned with the application of the knowledge in a manner that enables the design of either things or processes, and thus enables the satisfaction of needs. In contrast, evaluation is concerned with the capability that enables making judgments about what design is appropriate for a particular application. To demonstrate how these levels may be applied specifically to SE, Table 21 shows examples of various Bloom cognitive domain level competencies that might apply to GRCSE curricula and courses.

A discussion of the hierarchical nature of Bloom's taxonomy must be careful to clearly describe the nature of the levels in the hierarchy. Because Bloom's taxonomy is described in a hierarchical manner, it is possible to misunderstand the levels as being simply a higher level of attainment in a strictly quantitative sense. However, progression from one level to another is not only the result of more study, but also results from the direction of the study effort to develop a different kind of capability. For example, progression from “Knowledge” to “Comprehension” is not achieved by more study of the same kind that achieved the original knowledge, rather it is achieved by the study and performance of tasks that redirect the student from the kind of activities required to develop knowledge to the kind of activities required to develop comprehension. Similarly, “Analysis” and “Synthesis” are different kinds of skills that involve a different approach to thinking about the subject matter. A student needs to pass through the intellectual development stage of analysis in order to develop the ability to synthesize; synthesis is not simply more analysis, but rather a different kind of activity.

The educational goal is to raise the level of attainment of the student to the higher levels of Bloom's taxonomy, which will result in the student developing skills that enable that higher level of activity, even if the material content which the student has expressed at a higher level is of modest magnitude. This distinction is relevant in SE because the education program must develop people who can synthesize products and systems, and at later career stages, can evaluate proposals for systems. The kind of thought required to synthesize any product or system is similar, although the magnitude of attainment required increases with the magnitude and complexity of the system. Therefore, we may conclude that all systems engineers need the ability to synthesize systems, but that the magnitude of the system which their employer trusts them to synthesize will begin small and become larger as they demonstrate success.

Table 20. Explanation of Bloom Taxonomy Cognitive Levels (table developed for BKCASE). Source: (Bloom et al. 1956)

Level	Sub-Level	Competency	Outcome Descriptors

Knowledge (K)	• Knowledge of specifics
	  ◦ Knowledge of terminology
	  ◦ Knowledge of specific facts
	• Knowledge of ways and means of dealing with specifics
	• Knowledge of the universals and abstractions in a field
		Ability to remember previously learned material. Test observation and recall of information; i.e., “bring to mind the appropriate information;” e.g., dates, events, places, knowledge of major ideas, and mastery of subject matter.	List, define, tell, describe, identify, show, label, collect, examine, tabulate, quote, and name (who, when, where, etc.).

Comprehension (CO)	• Translation • Interpretation • Extrapolation	Ability to understand information and ability to grasp meaning of material presented; e.g., translate knowledge into new context, interpret facts, compare, contrast, order, group, infer causes, predict consequences, etc.	Summarize, describe, interpret, contrast, predict, associate, distinguish, estimate, differentiate, discuss, and extend.

Level	Sub-Level	Competency	Outcome Descriptors
Application (AP)	• Application of methods and tools • Use of common techniques and best practices	Ability to use learned material in new and concrete situations; e.g., use information, methods, concepts, and theories to solve problems requiring the skills or knowledge presented.	Apply, demonstrate, calculate, complete, illustrate, show, solve, examine, modify, relate, change, classify, experiment, and discover.
Analysis (AN)	• Analysis of elements • Analysis of relationships • Analysis of organizational principles	Ability to decompose learned material into constituent parts in order to understand structure of the whole. This includes seeing patterns, organization of parts, recognition of hidden meanings, and obviously, identification of parts.	Analyze, separate, order, explain, connect, classify, arrange, divide, compare, select, explain, and infer.
Synthesis (S)	• Production of a unique communication • Production of a plan, or proposed set of operations • Derivation of a set of abstract relations	Ability to put parts together to form a new whole. This involves the use of existing ideas to create new ones, generalizing from facts, relating knowledge from several areas, and predicting and drawing conclusions. It may also involve the adaptation of "general" solution principles to the embodiment of a specific problem.	Combine, integrate, modify, rearrange, substitute, plan, create, design, invent, what-if analysis, compose, formulate, prepare, generalize, and rewrite.
Evaluation (EV)	• Judgments in terms of internal evidence • Judgments in terms of external criteria	Ability to pass judgment on value of material within a given context or purpose. This involves making comparisons and discriminating between ideas, assessing the value of theories, making choices based on reasoned arguments, verifying the value of evidence, and recognizing subjectivity.	Assess, decide, rank, grade, test, measure, recommend, convince, select, judge, explain, discriminate, support, conclude, compare, and summarize.

Table 21. Example of Cognitive Levels for Systems Engineering (table developed for BKCASE).

Level	Example Competencies
Knowledge (K)	• The student is able to recite the definitions of "system" and "emergence" and state the connection between them. • The student is able to describe the notion of product system architecture and state the impact architecture may have on system development success.

Level	Example Competencies
Comprehension (CO)	• The student is able to explain, in a very general way, the conditions under which a system development team might choose to use a waterfall (or iterative, incremental, or spiral) life cycle model. • The student is able to explain the range of cases for which a particular systems modeling approach is applicable.
Application (AP)	• Given the operational concept and requirements of a simple system along with a specified budget and required completion time, the student is able to choose (and to provide a rudimentary justification for the choice) a particular life cycle model to address the project; e.g., waterfall, iterative, incremental, or spiral. • The student is able to construct a simple model of a defined system that would demonstrate understanding of the relationship of the primary factors included in the model.
Analysis (AN)	• Given a simple requirements document and a domain model for an application, the student is able to critique the domain model. • Given the operational concept of a system along with a requirements document, a budget, a schedule, a choice of a development process, and a justification of the use of that process for the project, the student is able to find and explain errors in the justification and/or in the choice of the process. • The student can analyze the effectiveness of a simple model of a system to describe the behavior of that system and identify errors or weaknesses in the model arising from the assumptions about the system embedded in the model.
Synthesis (S)	• Given a detailed requirements document and a well-constructed domain model for a system, the student is able to design at least one basic architecture for the system. • Given an operational concept, requirements, architecture, and detailed design documents for a system, the student is able to construct a complete implementation plan and provide a cogent argument that if the implementation of the architecture or detailed design is performed according to the plan, then the result will be a system that satisfies the requirements, fulfills the operational concept, and will be completed within the budget and schedule. • The student can develop and use a model of a simple system where the system is described by architecture to determine the capability of the system represented by the model and to explore desirable parameters of model elements.
Evaluation (EV)	• Given an operational concept, requirements, architecture, a detailed design, and an implementation plan, including budget and schedule, for a system, as well as a feasibility argument for the implementation plan, the student is able to assess the plan and to either explain why the feasibility argument is valid or why and where it is flawed with regard to any of the claims regarding implementation of the requirements, fulfillment of the operational concept, or the ability to be completed within budget and schedule. • Given a simple system, the student is able to plan a test and evaluation method to perform a verification and validation process of that system against the requirements of the system and the need description associated with the system. • Given a simple system and a test and evaluation plan of the system, the student is able to determine that the results that would be produced through use of the test and evaluation plan will yield a useful verification and validation of the system.

C.3 Bloom's Taxonomy Levels in the Affective Domain

Table 22 provides a description of the Bloom's Taxonomy for the affective domain. This information is included in GRCSE because the development of affective capabilities is part of what enables success in

SE (Sosniak 1994). To demonstrate how these levels may be applied specifically to SE, Table 23 shows some examples of various Bloom affective domain competencies that might apply to GRCSE curricula and courses.

Table 22. Explanation of Bloom Taxonomy Affective Levels (table developed for BKCASE). Source: (Krathwohl et. al. 1964)

Level	Sub-Level	Competency	Outcome Descriptors
Receiving (RC)	• Awareness • Willingness to receive • Controlled or selected attention	The learner is aware of stimuli and is willing to attend to them. The learner may be able to control attention to the stimuli.	Focuses on and is aware of aesthetics, focuses on human values, is alert to desirable qualities, and shows careful attendance to input.
Responding (RS)	• Acquiescence in responding • Willingness to respond • Satisfaction in response	The learner makes a conscious response to the stimuli related to the aesthetic or quality. At this level the learner expresses an interest in the aesthetic things.	Demonstrates willing compliance and obedience to regulations and rules, seeks broad-based information to act upon, and accepts responsibility and expresses pleasure for own situation.
Valuing (V)	• Acceptance of a value • Preference for a value • Commitment	The learner recognizes worth in the subject matter.	Continuing desire to achieve, assumes responsibility for, seeks to form a view on controversial matters, devotion to principles, and faith in effectiveness of reason.
Organization (OR)	• Conceptualization of a value • Organization of a value system	The learner is able to organize a number of values into a system of values and can determine the inter-relationships of the values.	Identifies characteristics of an aesthetic, forms value-based judgments, and weighs alternative policies.
Characterization (CH)	• Generalized set • Characterization	The learner acts consistently with the systems of attitudes and values they have developed. The values and views are integrated into a coherent worldview.	Readiness to revise judgment in light of evidence, judges problems and issues on their merit (not recited positions), and develops a consistent philosophy of life.

Table 23. Example Affective Levels for Systems Engineering (table Developed for BKCASE).

Level	Example Competencies
Receiving (RC)	• The student accepts that customer or user perception of the quality of a system is the fundamental determinant of system quality. • The student accepts that customers do not always fully describe what they want or need, and that there is a difference between what customers say they want and what they actually need. • The student is able to describe the value of the SE approach to design.
Responding (RS)	• The student learns how to ask questions to elicit the unstated desires of a stakeholder who is seeking a system development. • The student is willing to try the SE approach on a small project.
Valuing (V)	• The student believes it is important to provide system solutions that satisfy the range of stakeholder concerns in a manner that the stakeholders judge to be good. • The student believes it is important to elicit a nuanced description of what stakeholders desire of a system in order to provide rich knowledge that can be used in the system solution development. • The student believes in the value of the application of SE principles in a project, even in the face of advocates for other methods. • The student recognizes the value of advancing in proficiency in SE competencies.
Organization (OR)	• The student is able to organize a coherent framework of beliefs and understandings to support use of a SE method in a project. • The student has a coherent framework for how to discuss system development with stakeholders and to incorporate the views of a variety of stakeholders in a balanced manner.
Characterization (CH)	• The student will routinely approach system development projects with a SE framework. • The student will routinely evaluate the appropriate tailoring of SE processes to appropriately address the specific characteristics of each project. • The student will appropriately weigh the views of all stakeholders and seek to overcome conflicts between stakeholders using methods that are technically and socially appropriate.

This page intentionally left blank.

Appendix D: GRCSE Outcomes to CorBoK Mapping

Due to the recent completion of the *Core Body of Knowledge* (CorBoK), this appendix provides only notional information on how the *Graduate Reference Curriculum for Systems Engineering* (GRCSE) outcomes map to the CorBoK. Version 1.0 of GRCSE will provide the detail of the mapping of GRCSE outcomes to the CorBoK as one of several traceability items that will be completed (see Chapter 9 on SEBoK Evolution).

Chapter 3 describes the expected outcomes for the GRCSE framework. As defined in Chapter 3, expected outcomes delineate what students should be capable of performing when they graduate from a program that aligns with GRCSE's recommendations. Chapter 3 identifies 13 outcomes, which are repeated below. Chapter 6 defines the CorBoK for GRCSE. The CorBoK consists of foundation elements that define what *all* students should learn in a systems engineering (SE) master's program, and concentration elements, that define what students *expecting to assume a particular role* should learn in a master's program. The CorBoK structure is based on the *Systems Engineering Body of Knowledge* (SEBoK), which is organized into parts, knowledge areas (KA's), and topics. These elements are repeated in the CorBoK.

GRCSE outcomes:

1. ***Foundation*** – *Achieve designated Bloom's levels of attainment for each topic contained within the CorBoK foundation.*
2. ***Concentration*** – *Achieve designated Bloom's levels of attainment for each topic contained within one of the CorBoK concentrations, as appropriate for the type of master's program or for an individual student's interest.*
3. ***Topic Depth*** – *Achieve a Bloom's synthesis level of attainment for at least one topic from the CorBoK (either foundation or concentration).*
4. ***Application Domain*** – *Demonstrate the ability to perform SE activities in one application domain, such as defense, aerospace, finance, medical, transportation, or telecommunications.*
5. ***Specialty*** – The *Application of SE principles to address a specialty such as security, agility, or affordability*, *or such as safety-critical or embedded systems*.
6. ***Realism*** – *Comprehend and appreciate the challenges of applying* SE *to realistic problems throughout the system life cycle.*
7. ***Teamwork*** – *Perform as an effective member of a multi-disciplinary team, effectively communicate both orally and in writing, and lead in one area of system development, such as project management, requirements analysis, architecture, construction, or quality assurance.*
8. ***Problem/Solution Evaluation*** – *Be able to evaluate alternative system solution strategies, including how well different solutions relate to the identified problem, and express relevant criteria to ensure solutions are selected against a holistic systems perspective.*

9. ***Requirement Reconciliation*** – *Be able to reconcile conflicting requirements, finding acceptable compromises within limitations of cost, time, knowledge, risk, existing systems, and organizations.*
10. ***Professional Development*** – *Be able to learn new models, techniques, and technologies as they emerge, and appreciate the necessity of such continuing professional development.*
11. ***Related Disciplines*** – *Comprehend the relationships between* SE *and other disciplines, such as project management, human factors, and other engineering fields as discussed in the SEBoK, and be able to articulate the value proposition of these disciplines for* SE.
12. ***Software in Systems*** – *Demonstrate an understanding and appreciation of the level of software engineering necessary to develop current and future product*s, *service*s, *and enterprise systems.*
13. ***Ethics*** – *Demonstrate knowledge of professional ethics and of the application of professional ethics in decision-making and* SE *practice.*

Table 24 provides a notional mapping of the correlation of one of the GRCSE outcomes to the CorBoK KAs. The mapping of the CorBoK to the outcomes for outcomes 1 and 2 (listed above) are provided by the CorBoK tables provided in Chapter 6. Version 1.0 of GRCSE will provide traceability for each of the remaining outcomes for GRCSE.

The terms defining the degree of correlation between a CorBoK KA and an outcome are defined as follows:

- Strong: There is clear correlation between the achievement of a CorBoK KA and the achievement of an outcome; if the minimum Bloom's level in a KA is not attained by graduation, then there is a high probability that the corresponding outcome will not be achieved by graduation.
- Medium: There is a moderate correlation between the CorBoK KA and the specific outcome.
- Weak: There is marginal correlation between the CorBoK KA and the specific outcome.

Table 24. Notional Mapping of Outcomes to KAs in the CorBoK (table developed for BKCASE).

Knowledge Area	Outcome	
	SEM	SED
Part 2		
Systems Overview	Medium	Medium
System Concepts	Medium	Medium
Types of Systems	Medium	Medium
Representing Systems with Models	Medium	Medium
Systems Approach	Medium	Medium
Systems Challenges	Medium	Medium
Part 3		
Life Cycle Models	Weak	Weak
System Definition	Weak	Strong

Knowledge Area	Outcome	
	SEM	SED
System Realization	Weak	Strong
System Deployment and Use	Weak	Strong
Product and Service Life Management	Strong	Strong
SE Standards	Weak	Weak
Part 4		
Product SE	Strong	Strong
Service SE	Strong	Strong
Enterprise SE	Strong	Strong
Systems of Systems (SoS)	Strong	Strong
Part 5		
SE Organizational Strategy	Strong	Weak
Enabling Businesses and Enterprises to Perform SE	Strong	Weak
Enabling Teams to Perform SE	Strong	Weak
Enabling Individuals to Perform SE	Strong	Weak
Part 6		
SE and Hardware Engineering	Weak	Weak
SE and Software Engineering	Weak	Weak
SE and Project Management	Strong	Weak
SE and Specialty Engineering	Weak	Strong

This page intentionally left blank.

Appendix E: Assessment and Achievement of Learning Outcomes

This appendix provides additional background for Chapter 8. It describes a general framework for assessment that is independent of the domain covered by the program as well as its level (undergraduate or postgraduate). This framework is a generic reference that should be used to drive the definition and execution of assessment of actual systems engineering programs. This appendix does not present a prescriptive model, rather it offers guidance to help in building and applying assessment activities in concrete programs. Chapter 8 contains examples of applying that framework to assess systems engineering master's programs.

The primary purpose of assessment is to evaluate whether learning outcomes and objectives are being achieved. Assessment results should be independent of the people who perform them. An assessment should be as objective as possible, relying on well-defined, repeatable processes. An assessment should produce reliable and trustworthy results. It should offer a faithful view of reality without introducing biases that alter the result. Moreover, an assessment should be reproducible. Two assessments of the same objects should give similar results.

The activities students are asked to perform in a course, and in the program should correlate well with assessment criteria. Moreover, students should be clear on that correlation. For example, when a student group project is defined, normally one of the objectives is that students learn about group work. However, if there is no clear connection between how the project is assessed and the objective to learn about group work, students may not set out to develop and demonstrate their ability to perform well in groups, thereby missing a learning objective.

Some would argue that assessment of the deliverables of a project is the same as assessing the processes required to do the project. While it is clear that if the project is done in an appropriate way, the results should be good, it does not necessarily follow that if students produce a good product, they will necessarily conduct the project in a desirable way. Students' lack of experience with respect to appreciating project methodologies may lead them to follow an approach that has worked for other smaller tasks, and appears, given their lack of experience, to be appropriate, but actually may be inappropriate for a larger, more complex task. If the assessment explicitly looks at both the process and the product, and the student understands that they will be assessed on both, the student will likely focus on both.

E.1 What is the Purpose of Assessment?

According to ABET[12]:

> *assessment is one or more processes that identify, collect, and prepare data to evaluate the achievement of program outcomes and program educational objectives.* (ABET 2010, 1)

[12] The reference to ABET is used selectively here to support the definition of assessment term. As stated in the introduction to this appendix, it is very general and does not rely on any other existing framework. ABET is just one of the various sources of material and is not meant to be the sole reference for assessment.

More generally, an assessment can encompass a variety of objectives, from an individual student's achievement to the quality of an academic program. In detail, four levels of assessment are distinguished:

- Classroom assessment: involves assessment of individual students at the course level, typically by the class instructor.
- Course assessment: involves assessment of a specific course.
- Program assessment: involves assessment of academic and support programs. Program assessment is generally divided into two objectives:
    - Assessment of program objectives: what graduates are expected to attain within a few years of graduation.
    - Assessment of program outcomes: what students are expected to know and be able to do by the time of graduation.
- Institutional assessment: involves assessment of campus-wide characteristics and issues.

The logical relationships of the various targets of assessment are shown in Figure 9.

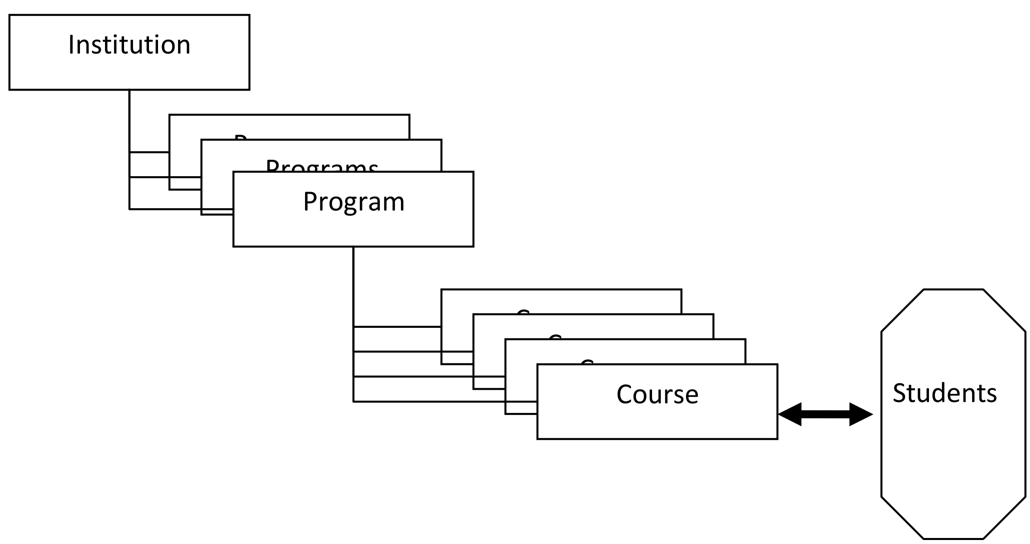

Figure 9. Various Targets for Assessment (figure developed for BKCASE).

These levels are not independent: assessing a program includes assessing the courses that constitute the program, which includes assessing students who are taking courses in the program. The achievement of program outcomes can be demonstrated by showing that students achieved the intended course objectives that are mapped to the program outcomes.

In the context of a graduate systems engineering program, the term "assessment" may encompass two kinds of objectives:

Summative: to ensure that the intended outcomes are achieved; to judge the quality of the students, course, and/or program.

Usually, summative assessments are formal assessments conducted at the end of lessons, projects, and/or courses to evaluate an individual student's learning achievement. Student summative assessments are graded and/or may lead to the selection and/or elimination of students.

Formative: to improve student competencies and course/program quality.

Formative assessments consist of sampling student learning and providing feedback to guide the learning process. In all instances, feedback rather than grading is the ultimate goal. Formative assessments give the instructor an opportunity to improve the learning experience and to modify the program or course teaching plan in order to improve the coverage of outcomes.

This appendix offers guidelines on how to define and conduct an assessment process that covers both student and program assessment, and potentially addresses both summative and formative purposes.

E.2 Context of Assessment

Assessment is a part of education. Gray (1977) described the educational paradigm in what he called the "*training triangle,*" which consisted of three points: outcomes, learning methods, and assessment, as illustrated in Figure 10. He showed that assessment was "*an integral part of the educational process*" as reported by the Merrison Committee (1975). It is the only way to measure whether students attain outcomes. Without assessment, students will not know how or whether they benefited from their training, teachers will have no idea whether the content of their programs is relevant or of the right quality, government or sponsor *organizations* cannot know whether their investments in universities and engineering schools are being spent wisely, and industry is unable to tell which emerging systems engineering graduates are competent.

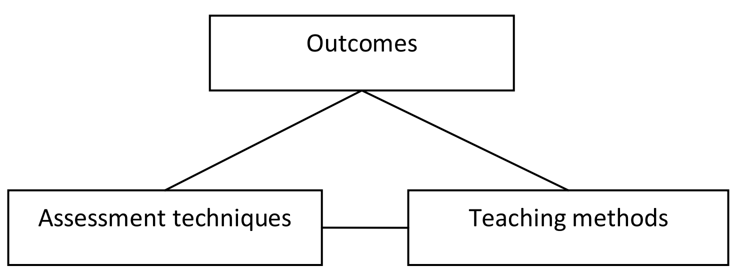

Figure 10. Training Triangle (figure developed for BKCASE). Source: (Gray 1977)

Outcomes, teaching methods, and assessment techniques should be connected very closely. The syllabus should clearly specify information about course learning outcomes, learning methods, and how learning assessments will be used.

E.3 Assessment Process

This section provides a general process to set up, lead, and exploit the results of assessment activities. This process is general enough to encompass program, course, and student assessment.

Program assessment is a way to continuously obtain feedback from students to guide the learning process. The assessment *life cycle* is a Plan, Do, Check, and Act (PDCA) process adapted from Chase and Aquilano (1995) and Wisconsin (rev. 2009):

- *Plan*: During this phase, the assessment plan is developed by identifying the course outcomes, defining the assessment strategy, and selecting methods to implement the assessment strategy. It is also during this phase that the organization of the assessment (tasks, who, When, effort, delays, etc.) and the infrastructure (tools supporting the assessment) are defined.

- *Do*: This phase involves teaching the program, setting up data and tools to perform the assessment, carrying out the assessment plan, and collecting/archiving assessment data.

- *Check*: During this phase, the results are analyzed and actions identified to improve the course. It consists of analyzing collected data and summarizing the results. The results and information gained should be distributed to students, faculty, and any other appropriate parties to obtain their ideas on how to improve.

- *Act*: During this step, course improvements are implemented and preparations are made for the following assessment cycle.

In this context, a general assessment process is represented in the following Figure 11.

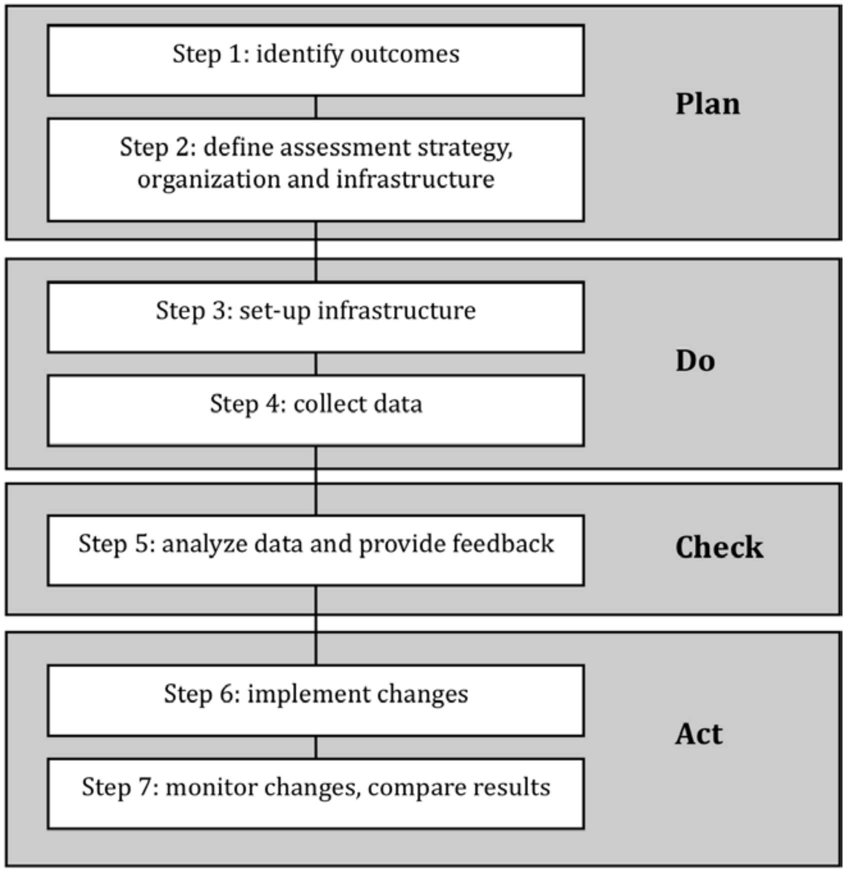

Figure 11. Assessment General Process (figure developed for BKCASE).

E.4 Detailed Assessment Process

Step 1: The objective of step 1 is to identify required outcomes: what the students are expected to know and be able to do during and at the end of the course or program.

Outcomes in GRCSE are specified using the identification of *knowledge areas (KAs)* or topics, and the corresponding levels of learning. GRCSE, as well as many instructors, use *Bloom's taxonomy* (Bloom et al. 1956) to describe levels for learning gains. More recent works can also be used (Gronlund 2000; Krathwohl 2002, 212-218).

Although GRCSE is mainly based on Bloom's taxonomy, it is helpful to highlight another well-known learning evaluation model provided by Kirkpatrick in 1959 (refer to (Kirkpatrick 1994) which included the original articles) where four levels of training assessment are defined. They represent a sequence of ways to evaluate programs. As assessment moves from one level to the next, the process becomes more time-consuming and difficult, but provides more valuable information:

Level 1: Reaction – measures the reaction and the satisfaction of the participants of the training program ("Are people happy with the training inputs?").

Level 2: Learning – measures the change in the participants' attitudes, or an increase of their *knowledge* as a result of the participation to the program ("What do people remember after the training session?").

Level 3: Behavior – measures how much knowledge is transferred, how much skill is gained, and whether ideal attitudes exist after training ("At work, do people use what they have learned?").

Level 4: Results – measures the global results and impact of the training on the business or environment over a period of time ("What are the outcomes of applications on the job?").

As with Bloom's model, Kirkpatrick's levels should be associated with outcomes. Outcomes should be assessed according to this classification using a dedicated assessment method. This model is now mainly used by industry where it is considered as a standard in the training community.

This first step should be recursive. A course takes place in a more general program and should be composed of individual lessons. After writing or analyzing course-level learning outcomes, the next step is to identify outcomes of the individual lessons (learning modules) that constitute the course. The lesson outcomes should clearly point back to the course objectives and support the achievement of course objectives. Course objectives should clearly point to program objectives and program objectives to the overall college or university's general education objectives.

Outcomes should be measurable (refer to Appendix D.4). Indeed, it should be possible to identify one or several performance criteria that correlate with the level of achievement of an outcome. Furthermore, this ability is important for graduation: it provides a capacity to determine if a student fulfills all program outcomes and is ready for a professional career. Performance criteria are compared to reference targets which must be achieved for graduation.

Step 2: Define assessment strategy and methods.

This step is at the heart of assessment because it is where the strategy and tactics of the assessment are defined.

Global objectives of assessment strategy should define:

- on which outcomes the emphasis of assessment should fall;
- how the assessment will be done (by selecting appropriate assessment methods);
- the organization of the assessment-identifying task, planning, and resources; and
- the infrastructure required to perform the assessment task.

Step 2.1: Define assessment methods.

The objective is to select assessment methods and to allocate them to each outcome to be assessed. Selecting the appropriate method for assessment is an essential step to ensure the success of the assessment process. A good assessment method should have the following characteristics:

- It should match the outcome. The selected assessment method should satisfy the objectives of the assessment questions. That is, the methods should be able to provide information about what is assessed according to the associated level. For example, the selected method should be adequate for the associated outcome Bloom's level.
- It should be reliable. A reliable assessment method yields consistent responses over time. To ensure that objective, the assessment should be reproducible: the same inputs should provide same results.
- It should be valid. Validity refers to determining whether the selected assessment method is appropriate for measuring what it is intended to measure. It is often a time-consuming and challenging task to provide evidence supporting the validity of the selected method. According to the Joint Committee on Standards for Educational Evaluation (Yarbrough et al. pending 2011), it is necessary to gather evidence to support the interpretation and appropriateness of a survey or test for a specific purpose. The use of multiple data sources is also recommended.

According to (Wiggins 1993), there is a trade-off between reliability and validity. The complexity of a task may increase validity, but at the same time, it will decrease reliability due to a lack of standardization. The key is to select methods that effectively balance the two issues according to the outcome level of learning. The assessment method should be adequate for the required level of learning associated with the outcome. This should also incorporate outcome priority. The assessment method(s) should be as precise as needed by the priority of the assessed outcome.

This information may be captured in a table indicating which outcomes each assessment method will measure. It is not necessary to assess all of the outcomes at once, but it is preferable to focus initially on those of the most importance or interest; i.e., the outcomes for the data that will be most useful to the program. Of course, one assessment method may measure multiple outcomes and one outcome may be measured by multiple assessment methods.

An additional consideration is the attractiveness of the assessment method. In some cases, an assessment method could be too complex or simply too long to be put in practice and this can lead to poor data; e.g., a complex or lengthy assessment survey. Such unattractive methods should be avoided.

Table 25 is an inventory of the major or most frequently used assessment methods (based on Fulks 2006). They have been categorized as either direct or indirect. Direct assessment methods focus on the competencies of students in the program and on their knowledge and skills. They are based on the examination of student performance through testing; review of student work, such as portfolios or theses; or observation of student performance in an appropriate context. Indirect methods differ because they deal with students' experiences, opinions, or perceptions. They can be self-reported by the student or can be obtained from faculty or employers. In general, direct measures are strongly preferred during assessment. However, direct measures may be impractical when assessing program objectives.

These two methods rely on feedback from diverse populations; e.g., internships, supervisors, student self-reports, employers, and teachers.

Table 25. Potential Methods of Assessment (Fulks 2004). Released by Janet Fulks, Assessing Student Learning in Community Colleges (2004), Bakersfield College.

Assessment Tool	Direct or Indirect	Formative or Summative	Bloom level	Kirkpatrick level		
Abbreviation	**D or I**	**F or S**	**K, C, A, A, S, E**		**Pros**	**Cons**
Multiple Choice Exam	D	F & **S**	K, C if carefully constructed A, S, & E	2	- easy to grade - objective	reduces assessment to multiple choice answers
Licensing Exams	D	S	K, C, A	2	easy to score and compare	no authentic testing, may outdate
Standardized Cognitive Tests	D	S	K, C, A?	2	comparable between students	heavily dependent on exposure to topics on test
Checklists	D	F, S	variable	2	- very useful for skills or performances - students know exactly what is missing	- can minimize large picture and interrelatedness - Evaluation feedback is basically a yes/no - present/absent - without detail
Essay	D	F, S	K, C, A, A, S, E	2, 3	displays analytical and synthetic thinking well	time consuming to grade, can be subjective
Case Study	D	F, S	K, C, A, A, S, E	2,3	- displays analytical and synthetic thinking well - connects other knowledge to topic	creating the case is time consuming, dependent on student knowledge form multiple areas
Problem Solving	D	F, S	K, C, A, A, S, E	2,3	- displays analytical and synthetic thinking well - authentic if real world situations are used	difficult to grade due to multiple methods and potential multiple solutions
Oral Speech	D	F, S	variable K, C, A, A, S, E	1,2,3,4	- easily graded with rubric - allows other students to see and learn what each student learned - connects general education goals with discipline-specific courses	- difficult for non-native language students - stressful for students - takes course time - must fairly grade course - content beyond delivery

Assessment Tool	Direct or Indirect	Formative or Summative	Bloom level	Kirkpatrick level		
Abbreviation	D or I	F or S	K, C, A, A, S, E		Pros	Cons
Debate	D	F, S	K, C, A, A, S, E	1,2,3	- provides immediate feedback to the student - reveals thinking and ability to respond based on background knowledge and critical thinking ability	- requires good rubric more than one evaluator is helpful - difficult for non-native language students - stressful for students - takes course time
Product Creation & Special Reports	D	F, S	variable K, C, A, A, S, E	3,4	students can display skills, knowledge, and abilities in a way that is suited to them	- must have clearly defined criteria and evaluative measures - "the look" cannot over-ride the content
Flowchart or Diagram	D	F, S	C, A, A, S, E	2,3	- displays original synthetic thinking on the part of the student - perhaps the best way to display overall high level thinking and articulation abilities	- more difficult to grade, requiring a checklist or rubric for a variety of different answers - difficult for some students to do on the spot
Portfolios	D	S	variable	2,3	- provides the students with a clear record of their work and growth - best evidence of growth and change over time - students can display skills. knowledge, and abilities in a way that is suited to them - promotes self-assessment	- time consuming to grade - different content in portfolio makes evaluating difficult and may require training - bulky to manage depending on size
Exit Surveys	D, I	S	A, S, E	2	- provides good summative data - easy to manage data if Likert-scaled responses are used	Likert scales limit feedback, open-ended responses are bulky to manage,
Performance	D	F, S	variable K, C, A, A, S, E	2,4	- provides best display of skills and abilities - provides excellent opportunity for peer review - students can display skills, knowledge, and abilities in a way that is suited to them	- stressful for students - may take course time - some students may take the evaluation very hard - evaluative statements must be carefully framed

Assessment Tool	Direct or Indirect	Formative or Summative	Bloom level	Kirkpatrick level		
Abbreviation	D or I	F or S	K, C, A, A, S, E		Pros	Cons
Capstone project or course	D	F, S	A,S, E	3,4	best method to measure growth overtime with regards to a course or program - cumulative	- focus and breadth of assessment are important - understanding all the variables to produce assessment results is also important - may result in additional course requirements - requires coordination and agreement on standards
Team Project	D	F, S	variable K, C, A, A, S, E	2, 3	connects general education goals with discipline-specific courses	- must fairly grade individuals as well as team - grading is slightly more complicated - student interaction may be a challenge
Reflective self-assessment essay	D, I	S	A, S, E	2,3,4	provides invaluable ability to evaluate affective growth in students	must use evidence to support conclusions, not just self-opinionated assessment
Satisfaction and Perception Surveys	I	S	C, A, A, S, E	1, 4	- provides good indirect data data can be compared longitudinally - can be used to determine outcomes over a long period of time	- respondents may be influenced by factors other than those being considered - validity and reliability most be closely watched

Most outcome assessment should be based on direct assessment; however, there are cases in which indirect assessments are needed for some outcomes, more often in relation to the soft skills. On the other hand, the majority of the objectives are best assessed with indirect assessment techniques: surveys, advisory committees, and interviews. However, there are potential direct assessment tools that can be used for some of the objective assessments.

For a systematic presentation and analysis of assessment methods, refer to (Prus 1994) or (UCF 2008).

Step 2.2: Define the life cycle of assessment.

The second step of the strategy defines when assessment is performed during the course. When performed at the beginning or at the middle of the course, assessment provides feedback that is directly usable by students and teachers while the course is on-going. In this case, assessment is clearly formative. When performed at the end of the course, assessment is a way to measure final learning achievement for grading purposes. It is also a way to improve the course. In this sense, final assessment can be both summative and formative.

Step 2.3: Define infrastructure to support the assessment process.

During this step, the tools and input data that are required to perform the assessment are specified. The required infrastructure depends on the assessment methods selected. For example, when a test is the selected assessment method, the definition of infrastructure would be the specification of data constituting the test, such as a set of test questions.

Step 2.4: Define the organization for assessment.

The objective of this step is to perform project management tasks related to the assessment process: define tasks, identify inputs and outputs for each task, identify planning, and forecast the workload necessary to perform the assessment tasks. Organization should also focus on the people implied in the assessment process: definition of roles, identification of people to support roles, and definition of recruitment and training needs.

It is important to perform assessment through diverse stakeholders involved in the training program: students, teachers, employers, and former students. This can take the form of a strategic advisory board. External assessors should be educated in systems engineering and should have a clear understanding of the approach the university takes. Diversity of stakeholders enables many viewpoints and a high capacity for curriculum evolution and development.

Step 3: Set up infrastructure.

This step encompasses the production of the infrastructure according to the specifications developed in step 2.3, as well as the realization or acquisition of material supporting the assessment, such as data and tools.

Step 4: Collect data.

This step consists in the execution of the assessment specified in step 2 using data and tools specified at step 2.3 and realized in step 3.

Step 5: Analyze collected data and provide feedback.

Collected data may be analyzed according to specific points of view:

- *Student selection*: In the case of summative assessment, data are used to identify a subset of students according to their level of learning achievement.

- *Student learning*: Data from measures associated with student learning outcomes permit the comparison of actual student performance with intended student performance, and identifies areas of strength and weakness for students, as well as ways to improvement the learning program.

- *Program and department processes*: Data from measures associated with processes related to the department and the program provide information that can improve the program's functioning and its efforts to facilitate students' progress toward graduation.

Data quality should be reviewed before it is analyzed to identify confounding factors indicated by the sources and the environment at the time of data collection. Identifying these factors can lead to modifying the assessment strategy; e.g., changing which data is collected as well as data collection methods (see step 6). For example, the sample size influences the validity of assessment findings and should be taken into account before actions are developed based on analysis results.

Step 6: Implement changes:

The following categories are areas, within the academic environment, where changes may be implemented.

Changes to assessment strategy may include:

- revision of assessment methods and their relationship to outcomes;
- collection and analysis of additional data and information; or
- changes to data collection methods.

Changes to program outcomes may include:

- revision of intended learning outcome statement(s) and their priorities.

Changes to *curriculum* may include:

- changes in learning methods;
- revision or enforcement of expected student backgrounds;
- revision of course sequences;
- revision of course content; and/or
- the addition or deletion of course(s).

Changes to academic processes may include:

- modification of the frequency or scheduling of course offerings;
- improvements in technology;

- changes in personnel;
- implementation of additional training;
- other implemented or planned changes;
- revision of advising standards or processes; and/or
- revision of *admission requirements*.

The length of time and degree of difficulty of these changes may vary greatly; e.g., revising admission criteria may require approval from multiple administrative levels within a university while changes in assessment data collection methods may be under the complete control of the faculty running the systems engineering graduate program.

Step 7: Monitor changes and compare results.

This is the last step of the continuous improvement cycle where planned changes should have been implemented. The implemented changes should be monitored to determine whether they have the desired effect(s).

One way of achieving this is to use the same assessment plan as used in the previous cycle and compare the actual data to the intended data. The drawback with this approach is that the people doing the assessment will not necessarily be the same each time. This introduces the possibility of changes induced by observer bias. However, this may also be an advantage when the new assessor is not influenced by past observations or relationships. It might also be worthwhile to reach out to the people who performed the former assessment, describe the changes implemented, and obtain feedback at least on the direction of the changes, even though these individuals may not have experienced the changes themselves.

Additionally, the environment in which the program operates changes over time, thus driving the program to adapt in order to continue to provide suitable education to the students and the community that the program serves.

Any gaps should be studied carefully to determine the underlying cause. Where outcomes have been met, the best course of action might be to continue monitoring the outcome to maintain quality.

E.5 On the "Good Definition" of Outcomes

For the purpose of assessment, outcomes must be clearly defined. The outcomes must be statements that can be measured so it is possible to determine whether or not students have achieved the intended goals. Well-defined outcomes are, in the context of assessment, similar to requirements in systems engineering practice; they should be SMART:

Specific: an outcome which is dedicated to the course. Specific outcomes should include in clear and definite terms the expected abilities, knowledge, values, and attitudes a student who successfully completes the course is expected to have. They should focus on intended outcomes that are critical. When the data from the assessment process are known, these outcomes should create opportunities to make improvements in the course that is being offered to students.

Measurable: it should be feasible to collect accurate and reliable data. Considering available resources (e.g., staff, technology, assessment support, and institutional level surveys) it should be possible to collect data for each student outcome.

Achievable: When defining the learning outcomes and setting targets, targets should be incremental; they should be designed to measure movement in the direction of a global objective, but not necessarily to measure whether that global objective has been achieved.

Results-oriented and *Time-bound*: When defining outcomes, it is important to describe performance targets: targets that are expected from a student in the course (e.g., 100% of graduates will achieve all outcomes and/or 80% of graduates will pass the written portion). Moreover, if previous measures of outcomes are available, it is helpful to use these as the baseline for setting a target for the next assessment.

Recall, outcomes should be measurable and achievable. It should be possible to identify for each outcome one or several performance criteria that indicate the level of achievement of an outcome by a student or a population. Furthermore, such performance criteria should be used to make comparisons to performance targets, to enable the possibility of establishing a relative classification between students, and to have an idea of the progress made by each student (if performance criteria are measured several times during a program or course duration).

E.6 Outcomes Classification

Outcomes may have varying levels of prioritization for a program. Some outcomes will be considered major knowledge targets for students while other outcomes may be defined as minor or less important.

An introduction of this classification is necessary because teaching and assessment strategies have to take into account opposing constraints: delivering the maximum value to students under limited resources (e.g., time, cost, and staff availability). Meeting time and cost constraints requires making decisions about which outcome(s) should be taught and assessed with a higher priority.

In the context of assessment, this classification is used as input for the strategy definition: assessment methods will be chosen according to the assessed outcome priority.

Outcomes classification is suggested by GRCSE in Chapter 7 that provides a set of classified SE outcomes through foundation and concentration topics. However, classification should be left to the responsibility of each university and school. This classification shall be done according to the general objectives of the program and/or to requirements produced by the appropriate authority in charge of awarding the academic degree.

Appendix F: Competency-Based Curriculum Development

F.1 Introduction

GRCSE™ provides the guidance necessary to create a systems engineering (SE) curriculum that supports attainment of the objectives described in Chapter 3 and the outcomes described in Chapter 4. The purpose of Appendix E is to provide recommendations and guidance on how to build a SE curriculum that will serve as a foundation for lifelong professional development. Developers of SE programs are encouraged to look beyond the academic environment and interact with industry, both to validate outcomes and objectives and to seek information and opinion about how their program might integrate with industry's approach to continuing development of their SE workforce.

In this appendix, the GRCSE authors discuss some of the basics of competency models used in classifying, assessing, and developing SE professionals. A process that uses competency model features to develop a GRCSE curriculum, which can better serve the professional development goals of an organization and the long-term career goals of graduates of a GRCSE-aligned program, is also described. Figure 12 is intended to capture the elements of the approach advocated in this appendix; these life-cycle elements are discussed further in the subsequent sections.

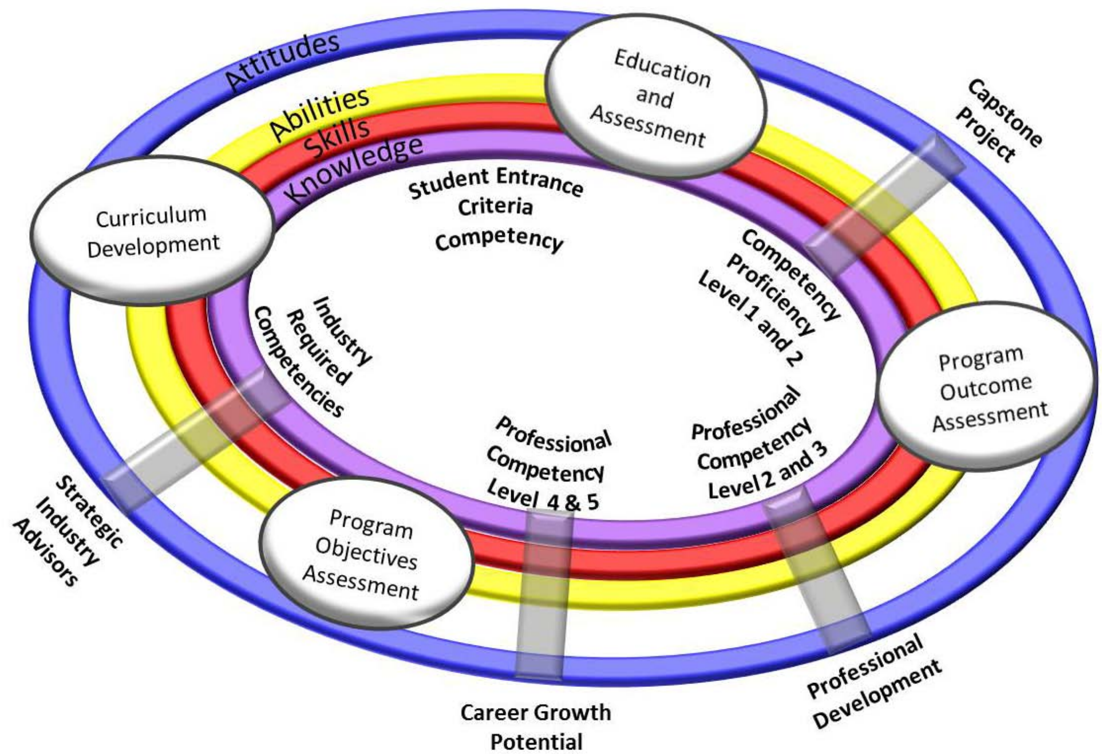

Figure 12. Competency Development Life-cycle (figure developed for BKCASE).

F.2 The Basics of Competency Models

The knowledge area (KA) within Part 5 of the SEBoK on "Enabling Individuals to Perform Systems Engineering" discusses the key elements of a competency model and provides examples of existing models developed by the US Department of Defense (DoD), the International Council on Systems Engineering United Kingdom chapter (INCOSE UK), MITRE, the US National Aeronautics and Space Administration (NASA), and the Software Engineering Institute (SEI). Ferris (Ferris 2010) summarizes and evaluates these models in terms of contribution to the selection of personnel and for informing systems engineering education.

The term "competency" is used in a number of different contexts. For example, in GRCSE we speak of outcomes, which are competencies of students upon graduation, and objectives, which are competencies students are expected to achieve three to five years after graduation. In this appendix, the term competency is meant to apply to a capability throughout the career of a systems engineer. The SEBoK defines competency as the set of knowledge, skills, abilities, and attitudes (sometimes referred to as KSAA) required for carrying out a job within an organization. Notice Figure 12 uses the KSAA elements to form guiding rings around the entire competency development life cycle with the curriculum elements distributed around the KSAA rings.

The following definition is more detailed: "a specific, identifiable, definable, and measurable knowledge, skill, ability, or other characteristic possessed by an individual necessary to the performance of an activity within an organization" (HR-XML Consortium 2004).

In developing a competency model for a large organization, hundreds of individual competencies may be identified. In order to organize the model and make it easy to use and understand, various approaches to grouping and classifying competencies have been developed. VanLeer (VanLeer 2007) classifies competencies as follows:

- *General Competencies* – company-wide competencies supported by all organizations, such as customer focus, communication, problem solving, systems thinking, and decision quality.
- *Organization Competencies* - system engineering competencies such as project management, requirements management, risk management, integration, and systems architecture.
- *Specializations* – competencies specific to a job type or an application domain (e.g., competencies specific to aviation systems).

Table 26 provides another view of how competencies may be categorized. Of course, both approaches could be used to organize and describe an organizations' competency model.

Table 26. Competency Categories (table developed for BKCASE).

Technical Competencies	Cognitive Competencies	Behavioral Competencies
Consists of specialized KSAAs specific to a domain — typically relates to hard skills and 'hands on' skills.	Are linked to cognitive-based activities such as problem solving, theory evaluation, and decision making.	Guides the interactions with others and includes attitudes towards and valuation of other competencies to guide appropriate use.
• Creating Requirements • Building an Architecture • Modeling System Performance	• Systems Thinking • Systems Performance Analysis • Trade-Off Analysis	• Business Ethics • Leadership • Conflict Resolution

The following is an example of how an organizational competency might be described:

> Requirements Management is the ability to analyze the stakeholder needs and expectations to establish and manage the requirements for a system.
>
> Key capabilities associated with the competency are:
>
> - Able to identify all the stakeholders and their sphere of influence.
> - Able to elicit requirements from stakeholders.
> - Able to create a traceability matrix of requirements for design and implementation.
> - Able to create a set of requirements that meets the quality specifications of writing a good requirement.
> - Able to define a process to manage the requirements and ensure their effective implementation.
> - Able to challenge the appropriateness of requirements in a rational way.

A requirements management competency could be classified using both the systems described above: it would best fall under the "organizational" competency classification, but aspects of "general" and "specializations" might apply; it also has elements of the "technical", "cognitive," and "behavior" categories.

Given a competency description, such as the one above, questions arise about what an organization's expectation is for different roles or responsibilities within this competency area. In order to determine the degree of competency needed for a particular job, and to assess whether an individual possesses a competency, models include a description of proficiently levels. Proficiency levels or levels of performance provide the measurement of the degree to which an individual possesses a competency. Appendix E describes the proficiency levels used in the competency model described in (VanLeer 2007) and Table 27 depicts another variation of proficiency levels described in (Squires et al. 2010).

Table 27. Competency Proficiency Levels (VanLeer 2007).
This information has been published with permission of Mary VanLeer.

Level 1: Absent	No knowledge and/or no experience with this competency.
Level 2: Conceptual	Basic training or learning has taken place around this competency.
Level 3: Applied	Repeated successful application of this competency. Help from an expert may be required.
Level 4: Expert	Can perform the actions associated with this competency without assistance. Notable strength.
Level 5: Innovator	Recognized visionary with this competency, able to formulate innovative solutions, new methods and standards.

Table 28. Competency Proficiency Levels (table developed for BKCASE).
Source: (APPEL 2009; Squires et al. 2010)

Level I: Participate (Know)	Performs fundamental and routine SE activities while supporting a Level II-IV systems engineer as a member of a project team.
Level II: Apply (Perform)	Performs SE activities for a subsystem or simple project (e.g., no more than two simple internal/external interfaces, simpler contracting processes, smaller team/budget, and shorter duration).
Level III: Manage (Lead)	Performs as a systems engineer lead for a complex project (e.g., several distinct subsystems or other defined services, capabilities, or products and their associated interfaces).
Level IV: Guide (Strategize)	Oversees SE activities for a program with several systems and/or establishes SE policies at the top organizational level.

Notice in Table 28 there are different levels of competency proficiency distributed throughout the competency development life cycle:

- At the lowest level there is the "Student Entrance Criteria Competency." This proficiency level is provided in GRCSE Chapter 4: Expected Student Background When Entering A Master's Program.
- Proficiency levels 1 and 2 (VanLeer 2007) are associated with the attainment of the student outcomes for a GRCSE curriculum. The capstone project is pictured in this part of the figure because this learning activity is typically used for outcomes assessment.
- Proficiency levels 2 and 3 (VanLeer 2007) overlap with attainment of the objectives associated with a GRCSE curriculum, which are achieved in the first years of systems engineering professional practice.

- Proficiency levels 3 and 4 (VanLeer 2007) are shown as part of continued career growth and may involve support from an industry advisor (mentor).
- Proficiency levels 4 and 5 represent reaching the top levels of the SE practice.
- The “Industry Required Competencies” speak to requirements of an organization that might influence all other proficiency levels.

F.3 A Competency-Based Curriculum Development Process

This section includes a process, shown in Table 29 that provides guidance on developing a competency-based curriculum. The intent is to use the GRCSE guidance provided in this document, along with an understanding and appreciation of professional career development, to develop a curriculum that recognizes competency development must extend throughout an individual’s career, and that competency models play a key part in assessing and advancing professional competency.

The general organization of the process described in Table 29 is based on a process in (Squires and Larson 2009) for assessing and improving an existing SE curriculum. Although the process in Table 29 is focused on the creation of new curriculum, with the elimination of Step 1, it could be adapted for use with existing curricula.

Steps 3 and 4 require activities beyond the basic GRCSE guidance. The papers by VanLeer (Van leer 2007) and Squires et al. (Squires and Larson 2009; Squires et al. 2010) offer ideas and guidance on establishing a competency model and on assessing gaps between actual and target proficiency levels.

Table 29. Process Script for Competency-based Curriculum Development (table developed for BKCASE). Source: (Squires and Larson 2009)

Process Step	Description
• **Step 1: Prepare Draft Curriculum**	• Establish program stakeholders. • Prepare draft outcomes and objectives. • Prepare draft curriculum.
• **Step 2: Establish the Development Team**	• Ideally, the team would be made up of faculty responsible for curriculum design and representatives from prospective employing organizations. • The team might use a wider group of reviewers for each stage of their work.
• **Step 3: Create Competency Model**	• Study and analyze various competency models. • Choose or adapt competencies appropriate to the draft curriculum and its stakeholders. • Define proficiency levels. • Select target proficiency levels for each competency.

Process Step	Description
• **Step 4: Assess Draft Curriculum against Competency Model**	• For each competency, identify where in the curriculum it is addressed and assess its summative proficiency level. • Aggregate and report on gaps between the “as is” and “to be” proficiency levels. Also, report on problems with the competency model (e.g., imprecise descriptions of proficiency levels).
• **Step 5*: Evolve Curriculum and Competency Model**	• Based on the competency assessment report, identify curriculum elements that appear weak (e.g., entrance expectations, outcomes, objectives, curriculum architecture, the core body of knowledge, or individual course activities. • Determine required changes in the curriculum and implement them. • Based on the competency assessment report, modify the competency model. • After implementation, repeat steps 4 and 5.

*Note: Step 5 is intentionally recursive – this ensures continual evolution and improvement of the curriculum.

Appendix G: Use Cases

It is the intention of the *Graduate Reference Curriculum for Systems Engineering* (GRCSE™) authors that GRCSE be useful to the global systems engineering (SE) educational community. In order to ensure this, the authors have developed a set of use cases for GRCSE. These use cases allow the GRCSE author team to identify gaps between what the community needs and the content GRCSE currently provides. The use cases also provide guidance to the community on the variety of ways that GRCSE content can support their needs. Additional discussion of implementation (not specific to a user) may be found in Chapter 7.

The current use cases (Version 0.5) are preliminary and present an outline of the information that will be provided in the final use cases. The final use cases (to be published in Version 1.0) will be fleshed out to include more narrative and specifics on how to implement GRCSE in certain situations. The GRCSE author team hopes that the reviewers of GRCSE 0.5 will provide feedback on

- whether the use cases listed are appropriate;
- whether there are additional use cases that should be added; and
- what information would be most useful for each use case.

For GRCSE 0.5, the authors have focused on five main use cases. These discuss utilizing GRCSE to support

1. SE Course development;
2. SE Curriculum development;
3. SE Curriculum assessment;
4. Non-SE Course development (incorporation of SE into courses for other engineering disciplines); and
5. Undergraduate and Graduate students.

For each use case, the primary goal is defined as well as the applicable user(s). If there are any specific assumptions for this use case, those are defined. Triggers that would cause the user to begin the activity are listed along with the expected process that would be followed. Finally, any assumptions regarding the outcomes of the use case are defined.

Use Case 1: Systems Engineering Course Development

Goal: A university faculty member or trainer uses the GRCSE document to design a course on SE.
User: University faculty member or a professional trainer.
Preconditions and Assumptions:

1. The user has identified areas of SE for which he/she plans to develop the course content.
2. The user has access to GRCSE.
3. The user has access to the *Systems Engineering Body of Knowledge* (SEBoK).
4. The user has a basic understanding of SE.
5. The user has searched for similar courses.
6. The user has searched for available text books on SE.
7. The user has identified an initial set of course objectives.
8. The user is familiar with the target student audience's educational and professional history.

Trigger: The user has identified a need for additional information about the scope of the course as it relates to the breadth and depth of coverage.
Normal flow:

1. The user consults the GRCSE document and reviews:
    a. GRCSE outcomes and objectives
    b. GRCSE *Core Body of Knowledge* (CorBoK)
    c. GRSCE CorBoK to SEBoK relationship
2. The user establishes an expanded set of outcomes and objectives for the course.
3. The user verifies the course outcomes and objectives against 1.a, 1.b, and 1.c.
4. The user makes appropriate modification to course outcomes and objectives.
5. The user proceeds to develop course artifacts (lecture material, references, exercises, exams, etc.).
6. The user identifies areas which require additional information.
7. The user refers to the SEBoK for additional information or references that point to additional information.
8. The user utilizes the SEBoK material and/or corresponding references to generate course artifacts.

Post-conditions:

1. The user has developed and packaged the necessary artifacts for the desired course.

Use Case 2: Systems Engineering Curriculum Development

Goal: A university faculty members or trainer uses the GRCSE document to design or modify the curriculum for a graduate program in SE.

User: A faculty member, group of faculty, and/or an academic administrator.

Preconditions and Assumptions:

1. The user has identified the need for the development of a new graduate program in SE or the modification of an existing program.
2. The user has reviewed some of the existing programs offered at other universities.
3. The user has access to GRCSE.
4. The user has access to the SEBoK.
5. At least one user has a deep knowledge of SE.
6. The user has identified the program constituents.
7. The user has identified the initial set of program objectives.

Trigger: The user has identified a need for the development/modification of graduate SE curriculum, and identified the need for reference curriculum.

Normal flow:

1. The user consults the GRCSE document and reviews:
    a. Entry requirements
    b. GRCSE outcomes and objectives
    c. GRCSE CorBoK and extended CorBoK
    d. GRSCE CorBoK to SEBoK relationship
2. The user establishes a set of outcomes and objectives or expands the set of existing outcomes and objectives for the program.
3. The user develops and verifies the curriculum outcomes and objectives.
4. The user makes appropriate modification to curriculum outcomes and objectives.
5. The user utilizes the CorBoK and CorBoK extension, and proceeds to design and develop the curriculum.
6. The user identifies areas which require additional information.
7. The user refers to the SEBoK for additional information or references that point to additional information.
8. The user utilizes the SEBoK material and/or corresponding references to generate a course description and syllabus.
9. The user utilizes Use Case 1, as appropriate, to develop or modify courses to support the curriculum.

Post-conditions:

1. The user has developed the desired graduate curriculum.

Use Case 3: System Engineering Curriculum Assessment

Goal: A faculty member uses the GRCSE document for the purpose of assessing the curriculum for a graduate program in SE.

User: Faculty, Student, Employer, etc. (For the purpose of this use case, this user is referred to as an evaluator, but this does not imply an "official" evaluator; e.g., an ABET evaluator.)

Preconditions and Assumptions:

1. The evaluator has identified the need for the assessment of a graduate program in SE.
2. The evaluator has access to GRCSE.
3. The evaluator has access to the SEBoK.
4. The evaluator has basic knowledge of the SE graduate curriculum to be evaluated.

Trigger: The evaluator has identified the need for assessing the graduate SE curriculum.

Normal flow:

1. The evaluator assesses the curriculum outcomes and objectives against GRCSE outcomes and objectives.
2. The evaluator reviews and assesses the curriculum content (courses) against the CorBoK and extended CorBoK.
3. The evaluator conducts gap analysis based on the results of the previous two steps.
4. The evaluator provides recommendations on (a) whether the gaps between the existing curriculum and GRCSE curriculum should be closed, (b) the rationale for this recommendation, and (c) a way ahead to use GRCSE to modify the curriculum as appropriate (see Use Case 1: Systems Engineering Course Development and Use Case 2: Systems Engineering Curriculum Development).

Post-conditions:

1. Gap analysis has been generated and assessment has been performed.

Use Case 4: Systems Engineering Course Development for Non-Systems Engineering Classes

Goal: A faculty member in an area outside of SE (e.g., electrical or mechanical engineering) uses GRCSE to identify SE content that could be included in his/her class(es).

User: A non-SE faculty member.

Preconditions and Assumptions:

1. The user has identified a need to include some SE content in his/her classes.
2. The user has access to GRCSE.
3. The user has access to the SEBoK.
4. The user is teaching in an engineering field.
5. The user has some basic understanding of issues in SE.
6. The user is familiar with the target student audience's educational and professional history.

Trigger: An engineering faculty member has identified a need for the inclusion of some SE content as part of their classes.

Normal flow:

1. A faculty member consults the GRCSE document and reviews:
    a. GRCSE Outcome and Objectives
    b. GRCSE CorBoK
    c. GRSCE CorBoK to SEBoK relationship
2. The faculty member establishes a set of SE outcomes and objectives to be included as part of his/her class(es).
3. The faculty member verifies the SE outcomes and objectives against 1.b, and 1.c.
4. The faculty member conducts research in the appropriate/identified areas of SE content that he/she wishes to include in his/her classes.
5. The faculty member proceeds to develop course artifacts (lecture material, references, exercises, exams, etc.).
6. The faculty member identifies areas which require additional information.
7. The faculty member refers to the SEBoK for additional information or the references within that point to additional information.
8. The faculty member feels comfortable with the material to be included in his/her class.
9. The faculty member uses the SEBoK material and/or corresponding references to generate course artifacts.

Post-conditions:

1. The user has included the necessary artifacts for the desired course.

Use Case 5: Graduate or Undergraduate Students

Goal: A graduate or undergraduate student uses the GRCSE document to learn about SE and/or pursue a graduate degree in SE.

User: A graduate or undergraduate student

Preconditions and Assumptions:

1. The user has interest
    a. in pursuing graduate study in SE and/or
    b. in evaluating available SE graduate degrees offered by a university or
    c. to learn about graduate SE education.
2. The user has access to GRCSE.

Trigger: The student has interest in pursuing graduate education in SE.

Normal flow:

1. Student consults the GRCSE document and reviews:
    a. GRCSE outcome and objectives
    b. GRCSE entrance requirements
    c. GRCSE CorBoK
    d. GRSCE CorBoK to SEBoK relationship
2. The student evaluates the content of CorBoK.
3. The student evaluates his/her preparation against entrance requirement.
4. The student confirms his/her interest in pursuing graduate degree in SE.
5. The student identifies potential schools offering graduate SE degree.
6. The student compares the candidate schools' curriculum against CorBoK.
7. The student identifies programs to which he/she would consider applying.

Post-conditions:

1. The user has learned about the graduate degree in SE.
2. The user identified gaps between his/her preparation and entrance requirement.
3. The user identified strengths and weaknesses of potential school's graduate SE curriculum (see Use Case 3: System Engineering Curriculum Assessment).