Graduate Reference Curriculum for Systems Engineering (GRCSE™)

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

GRCSE™ version 1.1 November 2015

Copyright© 2015 The Trustees of Stevens Institute of Technology on behalf of the BKCASE Board of Governors

BKCASE™ is supported by the following organisations

GRCSE Version 1.0 was developed under the leadership of

A compilation copyright to GRCSE™ and copyright to most of the content within GRCSE is held by The Trustees of the Stevens Institute of Technology 2015 ("Stevens") on behalf of the BKCASE™ Board of Governors. .

Prominently noted throughout GRCSE are other items of content for which the copyright is held by a third party. These items consist mainly of tables and figures. In each case of third party content, such content is used by Stevens with permission and its use by third parties is limited.

Stevens is publishing those portions of the GRCSE to which it holds copyright under a Creative Commons Attribution-NonCommercial ShareAlike 3.0 Unported License. See http://creativecommons.org/licenses/by-nc-sa/3.0/deed.en_US for details about what this license allows. This license does not permit use of third party material but gives rights to the public to freely use the remainder of GRCSE within the terms of the license. Stevens is publishing GRCSE as a compilation including the third party material under the terms of a Creative Commons Attribution-NonCommercial-NoDerivs 3.0 Unported (CC BY-NC-ND 3.0). See http://creativecommons.org/licenses/by-nc-nd/3.0/ for details about what this license allows. This license will permit very limited noncommercial use of the third party content included within GRCSE and only as part of the GRCSE compilation. Additionally, the U.S. government has limited data rights associated with GRCSE based on their support for GRCSE development.

Please cite GRCSE using the following information:
Pyster, A., D.H. Olwell, T.L.J. Ferris, N. Hutchison, S. Enck, J. Anthony, D. Henry and A, Squires (eds.). 2015. Graduate Reference Curriculum for Systems Engineering *(GRCSE™) V1.1. Hoboken, NJ, USA: Trustees of the Stevens Institute of Technology.*

Preface

The *Graduate Reference Curriculum for Systems Engineering* (GRCSE™) is a product of the *Body of Knowledge and Curriculum to Advance Systems Engineering* (BKCASE™) project. The original BKCASE project was led by Stevens Institute of Technology and the Naval Postgraduate School. This project created two products: GRCSE and the *Systems Engineering Body of Knowledge* (SEBoK) (BKCASE Editorial Board, 2015). SEBoK is a guide that provides an overview of key systems engineering (SE) knowledge areas and topics, and directs users to references from the SE literature. Version 1.0 of SEBoK was released in September 2012 and a number of updates have been made since. See www.SEBoKwiki.org for details of the latest version. GRCSE make significant use of SEBoK as a reference for the SE content element of its curriculum recommendations.

GRCSE Version 1.0 was released in December 2012. It was produced by the original BKCASE™ author team, with comments and contributions from across the Systems Engineering Community. The participation of professional societies in the creation of GRCSE was essential to ensure that it has the desired impact on global graduate education. The International Council on Systems Engineering (INCOSE), the Institute of Electrical and Electronics Engineers Computer Society (IEEE CS), the IEEE Systems Council, the Institute of Industrial Engineers (IIE) and the National Defense Industrial Association (NDIA) Systems Engineering Division provided representatives to serve as authors for the BKCASE project. The Association for Computing Machinery (ACM) provided review comments.

At the beginning of 2013, BKCASE transitioned to a new governance model with shared stewardship between the Systems Engineering Research Center (SERC), the International Council on Systems Engineering (INCOSE), and the Institute of Electrical and Electronics Engineers Computer Society (IEEE-CS). This governance structure was formalized in a memorandum of understanding between the three stewards that was finalized in spring of 2013. The stewards confirmed their commitment to making the SEBoK and GRCSE available at no cost to all users, a key principle of BKCASE.

The two BKCASE products are now managed by a BKCASE Editorial Board made up of volunteer editors and overseen by an appointed Editor-in-Chief. Please see www.bkcase.org for more information.

GRCSE Version 1.1 is a minor update produced by the BKCASE Editorial Board. A more significant review and update of GRCSE is planned for 2016.

GRCSE 1.0 Acknowledgments

The original BKCASE author team was led by Stevens Institute of Technology and the Naval Postgraduate School and composed of invited experts from industry, government, academia, and various professional associations. These authors followed an iterative, evolutionary approach when creating the two primary BKCASE products. The author team met in workshops approximately every three months from December 2009 to October 2012. GRCSE v0.25 was released in December 2010 to selected members of the systems engineering community with the invitation to review and provide the necessary feedback to develop subsequent versions. This feedback was received and the following year formed the basis of v0.5, which was released on a public website in December 2011, inviting review from anyone in the community. Version 1.0 is the result of further revision of 0.5; it is based on the review comments received from the community and further review and input the entire BKCASE author team.

Each BKCASE author, shown in Table 1 below, supported the writing of the SEBoK Version 1.0, GRCSE Version 1.0, or both. The 20 authors from 17 organizations who specifically supported GRCSE are shown in bold in Table 1. They came together selflessly and with the support of their organizations to improve global systems engineering education.

GRCSE Version 1.0 was edited by Art Pyster, David H. Olwell, Timothy L.J. Ferris, Nicole Hutchison, Stephanie Enck, James F. Anthony, Jr., Devanandham Henry, and Alice Squires. Special acknowledgement goes to Tim Ferris, the lead GRCSE author. Without his leadership and efforts, this document would not have been completed.

We gratefully acknowledge the strong support of the U.S. Department of Defense (DoD), which provided financial support for this project through the Systems Engineering Research Center (SERC) as well as providing 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 IEEE Systems Council, the National Defense Industrial Association (NDIA) Systems Engineering Division, the Institute for Industrial Engineers (IIE) 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, Justin Gercken, and Emily Leach, for their efforts that improved the readability and quality of GRCSE.

Table 1. Table of BKCASE Authors. Those in bold contributed to writing GRCSE. (GRCSE Original)

Author	Organization
Rick Adcock	**Cranfield University and INCOSE, UK**
James Anthony	Sevatec, Inc., USA
Erik Aslaksen	Sinclair Knight Merz, Australia
Richard Beasley	Rolls Royce, UK
Barry Boehm	University of Southern California, USA
John Brackett	**Boston University, USA**
Chuck Calvano	Naval Postgraduate School, USA

Author	Organization
Aaron Eng Seng Chia	**National 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
Leopoldo Decardenas	Raytheon, USA
Jeremy Dick	Integrated Systems Engineering Ltd., UK
Charles Dickerson	Loughborough University, UK
Dov Dori	Technion, Israel Institute of Technology, Israel and Massachusetts Institute of Technology, USA
Joseph J. Ekstrom	**Brigham Young University, USA**
Stephanie Enck	Naval Postgraduate School, USA
Marcia Enos	**Lockheed Martin, 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 and INCOSE, Australia**
Kevin Forsberg	OGR Systems, USA
G. Richard Freeman	US Air Force, USA
Sanford Friedenthal	SAF Consulting, USA
Brian Gallagher	CACI, USA
Devanandham Henry	Stevens Institute of Technology, USA
Michael Henshaw	Loughborough University, UK
Thomas Hilburn	**Embry-Riddle Aeronautical University, USA**
Nicole Hutchison	**Stevens Institute of Technology, USA**
Duane Hybertson	The MITRE Corporation, 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	The Aerospace Corporation, USA
Greg Mayhew	Boeing, USA
Steven Mitchell	**Lockheed Martin, USA**
Kenneth Nidiffer	Carnegie Mellon SEI and IEEE Systems Council, USA
David H. Olwell	**Naval Postgraduate School, USA**
Bo Oppenheim	Loyola Marymount University, 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

Author	Organization
Seiko Shirasaka	**Keio University, Japan**
Hillary Sillitto	Thales Group, UK
Janet Singer	International Society for the Systems Sciences, USA
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 Horta Travassos	**Universidade Federal do Rio de Janeiro (UFRJ), Brazil**
Ricardo Valerdi	University of Arizona, USA
Mary VanLeer	**The Sustainability Consortium, USA**
Qing Wang	Institute of Software Chinese Academy of Sciences, China
Brian Wells	Raytheon, USA
Brian White	Brian White, CAU<SES, USA
Ken Zemrowski	Ken Zemrowski, TASC, USA

In addition to the authors listed above, the following individuals have provided support to the BKCASE team over the course of the project. We gratefully acknowledge their assistance:

Johann Amsenga	JoAnn Lane	Peter Jackson
Johan Bendz	Sven-Olaf Schulze	Mike Kreuger
Dan Cernoch	Mary Jane Willshire	Richard Rosenthal
Edward Ghafari	Stuart Booth	Robert (Bob) Shishko
Richard Grzybowski	Richard Frost	
Kenneth Kepchar	Mike Gourley	

We also acknowledge the contributions of those who reviewed earlier versions of GRCSE. Their comments significantly improved the final product.

Table 2. GRCSE Reviewers 2010-2012. (GRCSE Original)

Name	Organization
Mark Ardis	Stevens Institute of Technology, USA
Vicenzo Arrichiello	SELEX Sistemi Integrati SpA, Italy
Radu F. Babiceanu	University of Arkansas at Little Rock, USA
Eric Bonjour	AFIS University of Franche Comte, France; University of Lorraine, France
Stephen Cook	University of South Australia, Australia
Rita Creel	Software Engineering Institute Carnegie Mellon, USA
Sharon Crossby	EADS Astrium, UK
Andrew Daw	Harmonic Limited, UK
Daniel J. Dechant	Raytheon Company, USA
Johnny D. Duckworth	Raytheon/Space & Airborne Systems, USA
Claudia A. Dunn	Raytheon, USA
Salah Eldin	North Carolina State University, USA
Erik Elich	Monto, Netherlands
Bruce Elliott	Arbutus Technical Consulting, UK
Susan Ferreira	The University of Texas at Arlington, USA

Name	Organization
Barthet Francois	Aeroconseil, France
Albert Gallo	US Department of Energy, USA
Tom Gilb	UK
Carol J. Gutierrez	Lockheed Martin Corporation, USA
Rick Hefner	Northrop Grumman Corporation, USA
Guilherme Horta Travassos	COPPE/UFRJ, Brazil
William Scott Jackson	University of South Australia, AUS
William J. Kennedy	Retired, fellow of IIE, professor emeritus of industrial engineering, USA
Adeel Khalid	Southern Polytechnic State University, USA
Suresh Krishnaswamy	IBM, India
Jeff Lankford	USA
Sarah E. Lewis	The Sustainability Consortium, USA
Jan de Liefde	Nspyre, Netherlands
Dahai Lui	Embry-Riddle Aeronautical University, USA
William Lyders	ASSETT, Inc., USA
Paola Di Maio	University of Strathclyde, UK
Linda Martinez	SYSTRA Engineering, Inc., USA
David Mason	Lockheed Martin, USA
Leon McGinnis	Massachusetts Institute of Technology (MIT), USA
Donald Medeiros	Raytheon Company, USA
Judy L. Miller	Charles Stark Draper Laboratories, USA
Adam Naramore	Alliant Techsystems Inc., USA
James A. Nemes	Penn State University, USA
David Newbern	TASC, Inc., USA
John B. Noblin	Lockheed Martin, USA
Mornas Odile	Thales, France
Edward Pines	New Mexico State University Industrial Engineering, USA
Ariel C. Pinto	Old Dominion University, USA
Valerie Rault	UDL INPL, ERPI/ENSGSI, France
Jack Ring	INCOSE, USA
Rich Rosenthal	TASC, Inc., USA
Theodora Saunders	Sikorsky, USA
Varaz Shahmirian	UCLA Extension, USA
Eric D. Smith	University of Texas at El Paso (UTEP), USA
Jim Smith	Lockheed Martin, USA
Richard S. Stansbury	Embry-Riddle Aeronautical University, USA
Dan C. Surber	Raytheon Technical Services Company, USA
Nora Tgavalekos	Raytheon, USA
Hans van Vliet	VU University, Amsterdam, Netherlands
Clayton Watkins	Beckman Coulter a Danaher company, USA
Thomas Yang	Embry-Riddle Aeronautical University, USA
Lori F. Zipes	NSWC PC, U.S. Navy, USA

With gratitude,

Art Pyster

BKCASE Editor-in-Chief and Project Leader 2009 to 2013

David H. Olwell

BKCASE Co-Editor-in-Chief and Co-Project Leader 2009 to 20013

Timothy L.J. Ferris, Editor and Lead Author
Nicole Hutchison, Editor
Stephanie Enck, Editor
James F. Anthony, Editor
Devanandham Henry, Editor
Alice Squires, Editor

GRCSE 1.1 Acknowledgements

GRCSE Version 1.1 has been prepared by the BKCASE Editorial Board. It includes minor changes to reflect both changes to the governance of the BKCASE project and recent updates to the SEBoK.

The only content change is to the GRCSE Core Body of Knowledge (CoRBoK) in Chapter 6. This has been updated to reflect the SEBoK Version 1.4 Knowledge Area and Topic headings. While the changes to SEBoK content in some areas have been significant, they do not affect the high level mapping of core knowledge areas in Chapter 6. Any program using the CoRBoK as part of its curriculum development or review will need to consider how best to follow the evolution of SEBoK detailed content as a part of their continuing review activities. Hence, this update does not represent a substantive change to the GRCSE guidance. A more significant update of GRCSE is planned for 2016, in which we will consider how to provide additional advice on how programs should keep themselves up to date with SEBoK updates.

Rick Adcock

BKCASE Editor in Chief

Table of Contents

PREFACE .... **III**

GRCSE 1.0 ACKNOWLEDGMENTS .... **IV**

GRCSE 1.1 ACKNOWLEDGEMENTS .... **VIII**

TABLE OF CONTENTS .... **IX**

LIST OF FIGURES .... **XI**

LIST OF TABLES .... **XI**

EXECUTIVE SUMMARY .... **XIII**

1. INTRODUCTION **1**
- 1.1 SYSTEMS ENGINEERING BACKGROUND 1
- 1.2 SYSTEMS ENGINEERING EDUCATION 1
- 1.3 DOCUMENT PURPOSE 2
- 1.4 GUIDANCE FOR DEVELOPING GRCSE 3
  - *1.4.1 GRCSE Scope 3*
  - *1.4.2 Global Applicability 4*
  - *1.4.3 Accounting for Multiple Education Pathways 4*
- 1.5 GRCSE OUTLINE 6

2. PROCESS FOR DEVELOPING OBJECTIVES **8**
- 2.1 OBJECTIVES 8
- 2.2 SETTING OBJECTIVES 9
- 2.3 EXAMPLE OBJECTIVES 10
- 2.4 GENERIC OBJECTIVES 12

3. EXPECTED OUTCOMES WHEN A STUDENT GRADUATES **14**
- 3.1 GRCSE EXPECTED OUTCOMES 14
- 3.2 GRCSE EXPECTED OUTCOMES 14
  - *3.2.1 Systems Engineering Concepts 15*
  - *3.2.2 Systems Engineering Role 17*
  - *3.2.3 Systems Engineering Practice 19*
  - *3.2.4 Systems Engineering Professionalism 20*
- 3.3 ALIGNMENT BETWEEN OUTCOMES AND OBJECTIVES 21

4.0 EXPECTED BACKGROUND FOR STUDENTS ENTERING A MASTER'S PROGRAM **23**
- 4.1 FINDINGS FROM THE SURVEY OF EXISTING PROGRAMS 23
- 4.2 GRCSE ENTRANCE EXPECTATIONS 24

5.0 CURRICULUM ARCHITECTURE **28**
- 5.1 INFLUENCES ON THE GRCSE CURRICULUM ARCHITECTURE 28
- 5.2 ELEMENTS OF THE GRCSE CURRICULUM ARCHITECTURE 28

6. THE CORE BODY OF KNOWLEDGE (CORBOK) **32**
- 6.1 CORBOK FEATURES 32
  - *6.1.1 CorBoK Foundation 32*
  - *6.1.2 50% Apportionment 33*
- 6.2 CORBOK ORGANIZATION 34

6.3 RECOMMENDED DISTRIBUTION OF CORBOK 50% 41
6.4 DEVELOPMENT OF THE CORBOK 41

7. IMPLEMENTATION 42
7.1 UNDERSTANDING INITIAL CONDITIONS 42
7.2 DEVELOPING OBJECTIVES 44
7.3 DEVELOPING OUTCOMES 44
7.4 ADDRESSING THE CORBOK 44
7.5 SETTING ENTRANCE REQUIREMENTS 45
7.6 ARCHITECTING COURSES AND CAPSTONES 45
7.7 DESIGNING THE ASSESSMENT PROCESS 46
7.8 CONTINUOUS IMPROVEMENT 46
7.9 EXAMPLES OF IMPLEMENTATION 46

8. ASSESSMENT 47
8.1 PROGRAM ASSESSMENT 47
8.2 ASSESSING PROGRAM OBJECTIVES 48
8.3 ASSESSING PROGRAM OUTCOMES 50
8.4 ASSESSING COURSE DELIVERY 53
8.5 STUDENT PERFORMANCE ASSESSMENT 55

9. EVOLUTION 58

GLOSSARY 60
LIST OF ACRONYMS 60
TERMINOLOGY 61

REFERENCES 67

APPENDIX A: CONTEXT AND GUIDANCE FOR THE CONSTRUCTION AND MAINTENANCE OF GRCSE 73
A.1 BACKGROUND 73
A.2 SYSTEMS ENGINEERING 73
A.3 ASSUMPTIONS 73
A.4 PRINCIPLES 74
A.4.1 Scope 75
A.4.2 Program Architecture 76
A.4.3 Context 77

APPENDIX B: SUMMARY OF GRADUATE SYSTEMS-CENTRIC SE PROGRAMS IN 2010 80
B.1 METHODOLOGY 80
B.2 FINDINGS 82
B.3 CONCLUSIONS 84

APPENDIX C: BLOOM'S TAXONOMY OF EDUCATIONAL OUTCOMES 85
C.1 INTRODUCTION TO BLOOM'S TAXONOMY 85
C.2 BLOOM'S TAXONOMY LEVELS IN THE COGNITIVE DOMAIN 88
C.3 BLOOM'S TAXONOMY LEVELS IN THE AFFECTIVE DOMAIN 92

APPENDIX D: GRCSE OUTCOMES TO CORBOK MAPPING 96

APPENDIX E: ASSESSMENT AND ACHIEVEMENT OF LEARNING OUTCOMES 98
E.1 PURPOSE, SCOPE AND JUSTIFICATION OF ASSESSMENT 98
E.2 ASSESSMENT PROCESS 100

APPENDIX F: COMPETENCY-BASED CURRICULUM DEVELOPMENT 107

F.1 INTRODUCTION ... 107
F.2 THE BASICS OF COMPETENCY MODELS ... 108
F.3 A COMPETENCY-BASED CURRICULUM DEVELOPMENT PROCESS ... 111

APPENDIX G: USE CASES ... 113
USE CASE 2: SYSTEMS ENGINEERING CURRICULUM DEVELOPMENT ... 116
USE CASE 3: SYSTEMS ENGINEERING CURRICULUM ASSESSMENT ... 117
USE CASE 4: SYSTEMS ENGINEERING COURSE DEVELOPMENT FOR NON-SYSTEMS ENGINEERING CLASSES ... 118
USE CASE 5: DEGREE SELECTION ... 118

APPENDIX H: EXAMPLE DISTRIBUTION OF TIME TO TEACH THE CORBOK ... 120
H.1 BACKGROUND ... 120
H.2 EXAMPLE DISTRIBUTION OF TIME DEVOTED TO THE CORBOK ... 120

List of Figures

FIGURE 1. THE DIVERSITY OF EDUCATION PATHWAYS ACCOMMODATED IN GRCSE. (GRCSE ORIGINAL) ... 4
FIGURE 2. GRCSE CURRICULUM ELEMENTS IN THE CONTEXT OF CURRICULUM DEVELOPMENT INFLUENCES. (GRCSE ORIGINAL) ... 6
FIGURE 3. THE RELATIONSHIP BETWEEN PROGRAM OBJECTIVES AND OUTCOMES, AND THE CORRESPONDING ASSESSMENTS. (GRCSE ORIGINAL) ... 8
FIGURE 4. ARCHITECTURAL STRUCTURE OF TOPICS IN A GRCSE MASTER'S PROGRAM. (GRCSE ORIGINAL) ... 30
FIGURE 5. COURSE ALIGNMENT WHICH MAY NOT DIRECTLY CORRESPOND TO TOPICS FROM ONLY ONE AREA. (GRCSE ORIGINAL) ... 31
FIGURE 6. CORBOK ORGANIZATION. (GRCSE ORIGINAL) ... 34
FIGURE 7. THE CHARACTERISTICS OF SYSTEMS ENGINEERING AS A PROFESSIONAL FIELD OF PRACTICE AND THE PLACE OF GRCSE IN SUPPORTING EDUCATION WHICH ENABLES THE CAPABILITIES REQUIRED. (GRCSE ORIGINAL) ... 86
FIGURE 8. VISUAL REPRESENTATION OF THE HIERARCHICAL NATURE OF BLOOM'S TAXONOMY IN THE COGNITIVE AND AFFECTIVE DOMAINS AND THE RELATIONSHIP OF THE LEVELS OF ACHIEVEMENT WITH COMPETENCY LEVELS. (GRCSE ORIGINAL) ... 87
FIGURE 9. VARIOUS TARGETS FOR ASSESSMENT. (GRCSE ORIGINAL) ... 99
FIGURE 10. TRAINING TRIANGLE (BASED ON PRINCIPLES FROM GRAY 1977). (GRCSE ORIGINAL) ... 99
FIGURE 11. ASSESSMENT GENERAL PROCESS. (GRCSE ORIGINAL) ... 100
FIGURE 12. COMPETENCY DEVELOPMENT LIFE-CYCLE. (GRCSE ORIGINAL) ... 107
FIGURE 13. GRCSE USE CASE DIAGRAM. (GRCSE ORIGINAL) ... 114

List of Tables

TABLE 1. TABLE OF BKCASE AUTHORS. THOSE IN BOLD CONTRIBUTED TO WRITING GRCSE. (GRCSE ORIGINAL) ... IV
TABLE 2. GRCSE REVIEWERS 2010-2012. (GRCSE ORIGINAL) ... VI
TABLE 3. OUTCOME TO OBJECTIVE RELATIONSHIPS. (GRCSE ORIGINAL) ... 22
TABLE 4. GRCSE PREPARATORY KNOWLEDGE. (GRCSE ORIGINAL) ... 25
TABLE 5. CORBOK BLOOM'S LEVELS FOR PART 2: FOUNDATIONS OF SE TOPICS. (GRCSE ORIGINAL) ... 36
TABLE 6. CORBOK BLOOM'S LEVELS FOR PART 3: SE AND MANAGEMENT. (GRCSE ORIGINAL) ... 37
TABLE 7. CORBOK BLOOM'S LEVELS FOR PART 4: APPLICATIONS OF SE. (GRCSE ORIGINAL) ... 38
TABLE 8. CORBOK BLOOM'S LEVELS FOR PART 5: ENABLING SE. (GRCSE ORIGINAL) ... 39
TABLE 9. CORBOK BLOOM'S LEVELS FOR PART 6: RELATED DISCIPLINES. (GRCSE ORIGINAL) ... 39
TABLE 10. . SYSTEMS ENGINEERING PROGRAM DEVELOPMENT PHASES MAPPED TO GRCSE (SQUIRES ET AL. 2011). RELEASED BY THE AMERICAN SOCIETY FOR ENGINEERING EDUCATION (ASEE). ... 42

Table 11. Example Rubric for Performance Measure for Outcome Assessment. This example uses a five-point Likert scale, with lower numbers being better. (GRCSE Original) 51
Table 12. Example of a Single Page Assessment Plan for an Outcome. (GRCSE Original) 52
Table 13. Possible Methods for Assessing Bloom's Levels of Achievement for Outcomes. (GRCSE Original) 53
Table 14. Example of a Teacher Online Survey. (GRCSE Original) 54
Table 15. Example of Student Online Survey Results – Extract. (GRCSE Original).....55
Table 16. Possible Methods for Assessing Bloom's Level of Achievement. (GRCSE Original) 56
Table 17. Survey Data Collection Instrument. (GRCSE Original).....81
Table 18. Survey Results. (GRCSE Original) 82
Table 19. Explanation of Bloom Taxonomy Cognitive Levels. (GRCSE Original) 89
Table 20. Example of Cognitive Levels for Systems Engineering. (GRCSE Original).....90
Table 21. Explanation of Bloom Taxonomy Affective Levels. (GRCSE Original) 93
Table 22. Example Affective Levels for SE. (GRCSE Original) 93
Table 23. Alignment of GRCSE Outcomes and the CorBoK. (GRCSE Original) 96
Table 24. Potential Methods of Assessment (Fulks 2004). Released by Janet Fulks, Assessing Student Learning in Community Colleges (2004), Bakersfield College. 102
Table 25. Competency Categories. (GRCSE Original) 109
Table 26. Competency Proficiency Levels (VanLeer 2007). Permission Granted by Mary VanLeer. All other rights are reserved by the copyright owner. 110
Table 27. Competency Proficiency Levels (Squires et al. 2010). (GRCSE Original) 110
Table 28. Process Script for Competency-based Curriculum Development. (GRCSE original) 111
Table 29. Process Script for Competency-Based Curriculum Development (Squires and Larson 2009). (GRCSE Original) 112
Table 30. Use Cases and the Users to Which They Apply. (GRCSE Original) 113
Table 31. Example Distribution of Time to The CorBoK. (GRCSE Original) 120

Executive Summary

The *Graduate Reference Curriculum for Systems Engineering* (GRCSE™) is a set of recommendations for the development and implementation of a systems-centric (Fabrycky 2010) professional master's degree program in systems engineering (SE). It is one of two products of, the Body of Knowledge and Curriculum to Advance Systems Engineering (BKCASE™) project. The other product, on which GRCSE draws, is the *Guide to the Systems Engineering Body of Knowledge* (SEBoK) (BKCASE Editorial Board, 2015).

Despite the importance of graduate education to meet industry needs for systems engineers, there has been no community-accepted recommendation or guidance on what to teach graduate students about SE. GRCSE offers guidance for systems-centric professional master's degree in SE, including:

- 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.
- A set of outcomes identifying the specific competencies that will be possessed by a student who successfully completes a graduate program based on the curriculum.
- A set of objectives aligned with those outcomes and describing the near-term career goals of such a graduate.
- Guidance on curriculum content, including a Core Body of Knowledge (CorBoK).
- A conceptual architecture to communicate and support implementation of the curriculum to achieve the outcomes and objectives.
- Guidance on program implementation and assessment, and how to balance the CoRBoK with the needs of the national or industrial domains being served.

The recommended topics provided by GRCSE are drawn from the SEBoK. A core subset of SEBoK topics was identified, referred to herein as the Core Body of Knowledge (CorBoK). This core consists of foundational topics that should be learned by all SE students, and an initial set of concentrations aimed at students focusing on a specific SE role such as system design and development or systems engineering management. GRCSE does not recommend a particular organization of these topics into courses. We expect that a program might dedicate up to 50% of the curriculum to core topics; with the rest covering select topics in more detail and domain-specific material that focuses a specific program on a target audience. How this is organized is deferred to the individual educational institution.

In addition to identifying specific topics to be covered, GRCSE includes a strong recommendation that a professional master's program in SE includes some sort of capstone experience. Systems engineering is by nature an intensely practical discipline; therefore, students must learn how to integrate theory and practice. An appropriate capstone experience forces such integration, while offering another dimension for tailoring the reference curriculum to the specific objectives of the university program.

GRCSE is designed for worldwide use. As such, it respects and accommodates the tremendous diversity in the markets that universities serve, the educational systems under which those universities operate, variations in size of student body and faculty, accreditation programs, and other factors that affect program content and delivery. Therefore, GRCSE provides a broad set of recommendations intended to guide universities in building and updating their graduate programs. It is not a standard, nor is it intended to be a foundational document for developing accreditation criteria to be used by accreditation agencies. GRCSE provides guidance to faculty members at educational institutions who are designing and updating graduate programs in SE, and to prospective students and their future employers to support the selection of a master's programs in SE that meets their particular needs.

Depending on the target market served by a particular program, prospective students may be mid-career practitioners with undergraduate degrees in SE or other fields of mathematics, science or engineering who are approaching a master's in SE as a career development step. Alternatively, prospective students may be recent graduates with little work experience, which presents a different set of challenges both for themselves and for the educational institution. The structure of GRCSE provides sufficient flexibility to accommodate both use cases, as well as many others.

The original BKCASE author team was led by Stevens Institute of Technology and the Naval Postgraduate School and was composed of invited experts from industry, government, academia, and various professional associations. These authors followed an iterative, evolutionary approach when creating the two primary BKCASE products. The BKCASE author team met in workshops approximately every three months from December 2009 to October 2012. GRCSE v0.25 was released in December 2010 to selected members of the systems engineering community with the invitation to review and provide the necessary feedback to develop subsequent versions. This feedback was received and the following year formed the basis of v0.5, which was released on a public website in December 2011, inviting review from anyone in the community. Version 1.0 is the result of further revision of v0.5, based on the review comments received and review work performed by the GRCSE author team, followed by review and input from and the entire BKCASE author team.

The participation of professional societies in the creation of GRCSE was essential to ensure that it has the desired impact on global graduate education. The International Council on Systems Engineering (INCOSE), the Institute of Electrical and Electronics Engineers Computer Society (IEEE CS), the IEEE Systems Council, the Institute of Industrial Engineers (IIE) and the National Defense Industrial Association (NDIA) Systems Engineering Division provided representatives to serve as authors for the BKCASE project. The Association for Computing Machinery (ACM) provided review comments.

At the beginning of 2013, BKCASE transitioned to a new governance model with shared stewardship between the Systems Engineering Research Center (SERC), the International Council on Systems Engineering (INCOSE), and the Institute of Electrical and Electronics Engineers Computer Society (IEEE-CS). This governance structure was formalized in a memorandum of understanding between the three stewards that was finalized in spring of 2013. The stewards have reconfirmed their commitment to making the SEBoK available at no cost to all users, a key principle of BKCASE.

The two BKCASE products are now managed by a BKCASE Editorial Board made up of volunteer editors and overseen by an appointed Editor-in-Chief. Please see www.bkcase.org for more information.

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. (INCOSE 2012)

SE principles and practices are essential for the development of large, complex, and/or trustworthy systems, whether they are products, services, or enterprises. SE enables a contemporary lifestyle that relies on high-performance systems in every sector, and affects all aspects of life. Modern design philosophies demand systems thinking in order to support the deep integration of technical systems and organizations required to supply the kinds of services now expected. SE helps ensure that the system delivered is a coherent and effective solution to the system need.

Systems engineers are responsible for ensuring that the right system development tasks are performed so that high-quality systems can be both delivered and sustained. Systems engineers work with a variety of other professionals to create and support systems throughout their life cycle. These other professionals contribute skills in various engineering disciplines, systems science, and project management, among many other fields. 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 combining these into a coherent representation of the system requirements.
- Understanding the technology that may be applied in the system.
- Appreciating the life cycle implications of systems and incorporating life cycle perspectives into systems design.
- Evaluating, selecting, and developing system solutions to satisfy customer needs and project objectives.

The knowledge, skills, abilities, and attitudes (KSAAs) 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 KSAAs are becoming critically important. Such curricula should enable students to continue their professional development after graduation and eventually to move into engineering and technical leadership of large, complex projects.

1.2 Systems Engineering Education

The roots of SE education extend 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, covering a general description of SE programs and a set of the technical issues confronting engineers at the system level. (Wymore 1962)

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 refers to these as "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. The majority of 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; with 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; 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[1] for a professional master's degree in SE to fill this gap.

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 2007, the United States Department of Defense's (DoD) Office of the Secretary of Defense (OSD) 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. The prior work has significantly informed the present work, GRCSE.

[1] 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. It is not a standard, nor is it intended to be a foundational document for developing accreditation criteria to be used by accreditation agencies.

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 implementation of those guidelines.
- Provide a framework to guide the deliberations of strategic advisory boards, established to assist universities in appropriate program design.
- Support increased enrollment in SE programs by increasing the value of those programs to potential students and employers.
- 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.
- Provide a presentation of systems engineering curriculum that will assist engineering educators in general to appreciate the distinctive knowledge and perspectives of systems engineering.

1.4 Guidance for Developing GRCSE

The GRCSE team began by stating principles and assumptions to guide the work. These are briefly summarized:

- SE is a distinct discipline and contains a rich body of knowledge. Therefore the SEBoK serves as an important input to GRCSE (BKCASE Editorial Board, 2015).
- SE interacts with other disciplines, some of which provide important foundational 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 objectives and outcomes sections of GRCSE (Chapters2 and 3, respectively).

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, universities may include content for purposes beyond those for which GRCSE is specifically intended. GRCSE does not address such extensions, as they are considered out of its scope.

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. Regulations and culture may:

- Constrain the nature of programs that can be offered at the master's level or course offerings or some other aspect of programs.
- Influence students' decisions as to whether to proceed directly from undergraduate to master's level studies or to gain some career experience between the two education stages.
- Influence the choice and acceptability of face-to-face, online, or hybrid delivery.
- Affect the administrative decomposition of programs.

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 1.

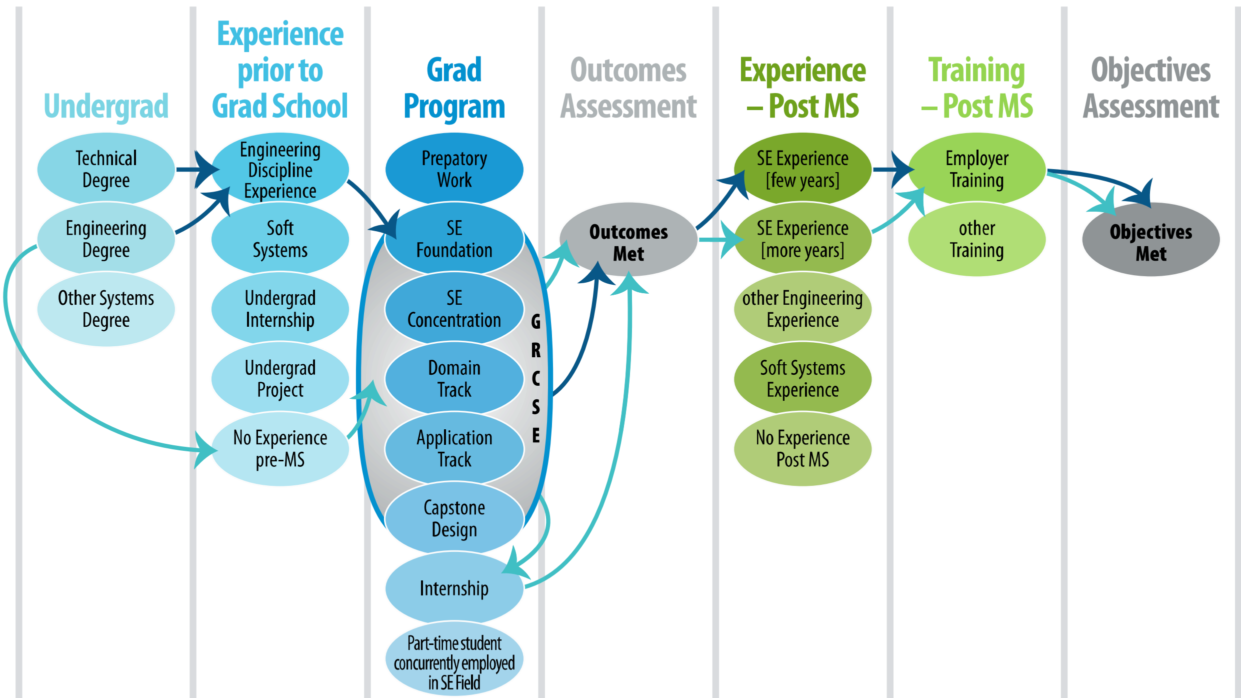


Figure 1. The Diversity of Education Pathways Accommodated in GRCSE. (GRCSE Original)

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 light blue in Figure 1). On the second path – which aligns with the GRCSE recommendations – the student gains at least two years of engineering experience between the bachelor and master’s degree level studies (shown in dark blue in Figure 1)? As stated in Chapter 4, at least two years of professional experience is recommended. Whether the student obtains work experience between completing a bachelor’s program and commencing master’s level studies is 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 light and dark blue paths. In the US, most students follow the dark blue path: they take jobs after their undergraduate education before pursuing a graduate degree in SE. In Europe, most students follow the light blue 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 may 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 1.0 recommendations address students following the dark blue path. If a program permits students with little experience in the workplace to enter, it is necessary to evaluate the impact of that gap in the student background and alter the implementation of the program to ensure that all students who enter can meet the outcomes and objectives of the program.

1.5 GRCSE Outline

Curriculum design addresses the desires and needs of students and potential employers along with university-specific requirements, objectives, and mandates, as shown in Figure 2 below. These contextual factors provide input to objectives, which are expectations of what a student will have achieved by a certain 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 and their organization that must be covered for students to achieve the outcomes. Outcomes must be assessed, and the assessments inform the on-going design of the overarching curriculum. Meanwhile, the available pool of potential students informs the program's expectations about student backgrounds at entry. Student ability to achieve outcomes partly depends on background; understanding this informs the program design.

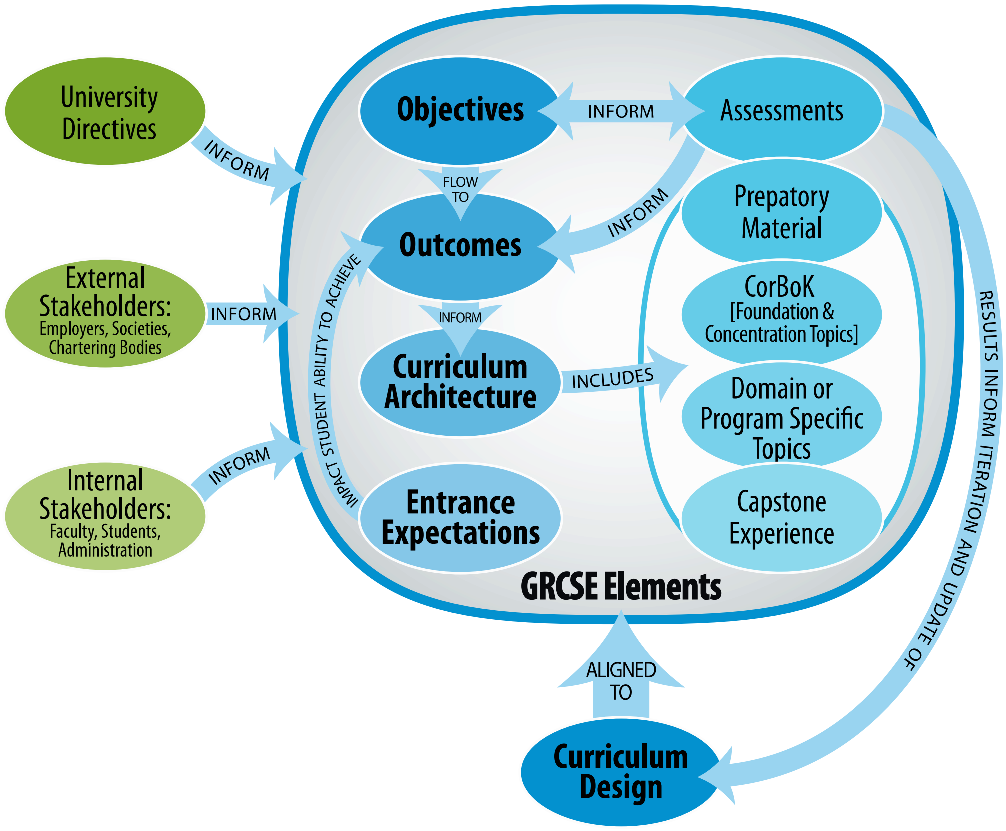


Figure 2. GRCSE Curriculum Elements in the Context of Curriculum Development Influences. (GRCSE Original)

All of these interactions are important aspects of curriculum design. GRCSE specifically focuses on the areas contained within the blue of Figure 2, labeled GRCSE elements. This information is presented in the following chapters:

- Chapter 2 - Describes the process for setting objectives – student attainment three to five years after graduation – as well as some example objectives.
- Chapter 3 - Examines 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 the recommended curriculum architecture for structuring a SE program and a common mechanism for communicating the components of an institution's SE graduate level curriculum.
- Chapter 6 - Describes the CorBoK, is organized according to the curriculum architecture described in Chapter 5, and defines expected levels of achievement of learning.
- 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.
- Chapter 8 - Provides guidance for developing assessment rubrics to ensure that graduate programs achieve their intended outcomes.
- Chapter 9 - Explains the intended evolution and long-term support of GRCSE.

Together, these chapters and supporting materials provided in the appendices 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[2].

[2] 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 for curricula that align with GRCSE recommendations. According to the Accreditation Board for Engineering and Technology (ABET[3]):

> *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, which:

> *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.* (ABET Engineering Accreditation Commission 2010)

Program outcomes are set by the educational institution in order 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 3.

Figure 3. The Relationship Between Program Objectives and Outcomes, and the Corresponding Assessments. (GRCSE Original)

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

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

engineering (SE) increases its maturity as an engineering discipline, defining its curricular design in the 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 employers of 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 3 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 B) 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 supplier.

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 KSAAs desired of the various classes of employees at different stages in their careers. These KSAAs 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 KSAAs. Workforce development plans use the competency models to map the desired KSAAs against the means used by the employees to obtain them.

Employees are often certified, either internally or externally, or licensed. In particular, the International Council on Systems Engineering (INCOSE) has a certification program for systems engineers. Certification or licensing serves as a designation that the engineers possess KSAAs to perform certain duties in the organization. If a program intends to channel students into a certification or licensure regimen, then 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; 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 (UK), 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.

All of these – competency models, certification regimes, licensure, accreditation, advisory boards – influence the design of program objectives.

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 and that 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's program 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 aeronautics (e.g., European Aeronautical Defence and Space (EADS), Airbus, and Air France), in electronics (e.g., Thales, Rockwell Collins, etc.), in software engineering (e.g., IBM, Steria, etc.), in space (e.g., Centre National d'Etudes Spatiales (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, in its various departments, holding similar positions (e.g., engineering, operation, research, teaching, and management).

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. The three to five year time frame is short enough to be practical for use in program maintenance and to avoid the influence of non-program factors in graduate progression, but also long enough for graduates to build the foundational experience of their careers.

Three to five years after graduation, program graduates are expect to be able to:

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 expert and leadership 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 be likely to accomplish in the near to mid-term in their career (see expectation 1 above) and to parallel those found in other engineering disciplines (see expectations 2, 3, and 4 above). Schools should tailor the objectives to their own program stakeholders.

Sheard (1996, 2000) describes twelve roles for systems engineers. A program and its stakeholders might select some of those roles explicitly when tailoring the second generic objective.

Academic institutions and employers might also consider intentional collaboration in both near-term cross-sectional studies, and longer-term longitudinal studies to investigate the effectiveness and usefulness of the SE education objectives. These studies could be done in cooperation with events such as the Conference on Systems Engineering Research (CSER) or with the university's engineering education and systems engineering faculty 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, in regard to 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 that is in accordance with 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

Outcomes 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.

Chapter 2 described the process for setting objectives. Building on that process, this chapter describes expected outcomes for programs aligned with the GRCSE recommendations. In addition, this chapter presents the relationships between the program outcomes and objectives, and how these outcomes support the achievements of the program objectives three to five years after graduation.

Appendix D represents a mapping between the program outcomes and the content of Core Body of Knowledge (CorBoK), and how the CorBoK content facilitates the attainment of the program outcomes.

3.1 GRCSE Expected Outcomes

The expected outcomes describe what students are expected to know and be able to do by the time of graduation to commence or enhance professional practice as a systems engineer. They include a mix of technical and professional capabilities that reflects the diverse skills that successful systems engineers require. Development of the outcomes was supported in part by the survey of graduate programs conducted in 2010 (see Appendix B for additional information).

The outcomes in this chapter are starting points from which a university should define its own set. Some outcomes, in particular outcomes related to the FOUNDATION, CONCENTRATION, and TOPIC DEPTH, recommend mastery at certain levels of Bloom's taxonomy (described in Appendix C). These can be associated with specific, measurable KSAAs, which are very useful for assessment. Associating similar measures with the more qualitative outcomes is more difficult. Where appropriate, some possible approaches for these are included.

GRCSE asserts that certain knowledge areas and topics in the Guide to the Systems Engineering Body of Knowledge (SEBoK) need to be covered in a graduate curriculum (BKCASE Editorial Board, 2015). These topics are called the core body of knowledge, or CorBoK. The Foundation and Concentration outcomes specifically describe the level of mastery expected for the CorBoK, which is discussed in detail in Chapter 6.

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 3, which is located at the conclusion of this chapter.

3.2 GRCSE Expected Outcomes

GRCSE program outcomes are divided to four major categories:

- SE Concepts,
- SE Role,

- SE Practice, and
- SE Professionalism.

Each category has three or four outcomes; the order does not imply a ranking. GRCSE presents the set of outcomes with the view that they all interact to form a coherent view of a program.

Upon graduation, students who attend a master's program in SE that follows the GRCSE recommendations are expected to satisfy all of the outcomes. These outcomes are described below (please note that the order indicates the order in which outcomes are discussed and does not imply rank or importance):

- *FOUNDATION* (SE Concepts)
- *CONCENTRATION* (SE Concepts)
- *TOPIC DEPTH* (SE Concepts)
- *APPLICATION DOMAIN* (SE Role)
- *SPECIALTY* (SE Role)
- *RELATED DISCIPLINES* (SE Role)
- *SOFTWARE IN SYSTEMS* (SE Role)
- *REQUIREMENT RECONCILIATION* (SE Practice)
- *PROBLEM/SOLUTION EVALUATION* (SE Practice)
- *REALISM* (SE Practice)
- *PROFESSIONAL DEVELOPMENT* (SE Professionalism)
- *TEAMWORK* (SE Professionalism)
- *ETHICS* (SE Professionalism)

These are described in detail in the following sections. As these outcomes are referenced throughout GRCSE, they are listed by their abbreviated name and formatted as shown above (e.g. "the *FOUNDATION* outcome"). Note that for each outcome above the category to which it belongs is listed in parentheses.

3.2.1 Systems Engineering Concepts

This group of outcomes provides the necessary breadth and depth of the knowledge required for all SE graduates to become contributing members of the profession. The SE Concepts category includes: *FOUNDATION*, *CONCENTRATION*, and *TOPIC DEPTH*.

- FOUNDATION – *Achieve the 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 includes 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 graduate 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 curriculum design and assessment processes. Appendix E offers some general guidance on the kinds of specific learning outcomes and associated assessment approaches that may be used to assess student achievement. Chapter 8 provides actual examples of outcomes and objectives developed as part of the assessment process.

- *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 roles for 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, they may need a different set of additional skills for their SE specialization. Each CorBoK concentration specifies additional topics and/or increased Bloom's levels for some topics, which will apply to different types of master's programs. Two concentrations are identified in version 1.0 of GRCSE: Systems Engineering Management (SEM) and Systems Design and Development (SDD).

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

- *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 6); 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 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, capstone design and/or projects. It is more likely that areas of depth will be taken from the CorBoK concentrations that are relevant to a program. However, the demonstration of synthesis-level ability in any 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.

3.2.2 Systems Engineering Role

Systems engineers are required to participate as productive members of interdisciplinary teams. They perform different tasks within different domains. The four outcomes in the SE Role category - *APPLICATION DOMAIN*, *SPECIALTY*, *RELATED DISCIPLINE*, and *SOFTWARE IN SYSTEMS* - provide the necessary background for graduates to perform effectively in these interdisciplinary environments.

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

  This 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. It incorporates an understanding of specialized terminology, technology, methods, tools, and constraints that are unique to the chosen application domain.

  This 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 the characteristics and peculiarities of that domain on a daily basis. Priorities, vocabulary, paradigms, technologies, tools, and a myriad of other factors vary from domain to domain; 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 in at least one domain through prior work experience, projects, and the capstone experience.

- *SPECIALTY – Apply SE principles in order to address a specialty, such as: security, safety, affordability, or safety-critical or embedded systems*.

  This outcome includes understanding how differences in specialties manifest themselves in both engineering a system and in the function of the system itself. It also includes the need for an examination of the skills that are 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; however, for an automobile braking system, safety is likely the most important property in developing a solution. Development standards are of critical importance in defense applications, but less important in systems used to create special effects in movies. It should be noted that 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 specialty is not emphasized in a given domain. Coupling the two may provide students with the opportunity to delve into each more deeply; if the two are not coupled, students may have more difficulty allocating appropriate time to each.

- *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 traditional engineering disciplines (electrical, mechanical, civil, etc.) as well as more management-focused disciplines (project management, program management, industrial engineering, etc.). It is important that systems engineers possess basic knowledge related to these disciplines and also understand how SE 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 (BKCASE Editorial Board, 2015).

  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 system success or failure.

- *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 effective SwE may fundamentally change the way SE is performed. For example, SwE understanding may open additional options for allocating functionality throughout a system. Students do not need to become experts in SwE, but should understand the unique characteristics of software to better enable them to support the development of software-intensive systems.

3.2.3 Systems Engineering Practice

There are number of phases, processes, activities, and methodologies involved in the development of a system from inception to disposal. The three outcomes in the SE Practice category provide the necessary background for graduates to successfully develop a reliable system: *Requirement Reconciliation*, *Problem/Solution Evaluation*, and *Realism*.

- *Requirement Reconciliation – Master the quantitative skills 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 means 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 some experience applying these techniques.

  Please note that there is a relationship between this outcome and the *Ethics* outcome ("*SE Professionalism*" below), as difficulties in reconciling conflicts within policy and legal guidelines can lead to ethical dilemmas.

- *Problem/Solution Evaluation – Master the quantitative skills to evaluate alternative system solution strategies, including how well different solutions relate to the identified problem, and express the 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, including: 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. For instance, centralized vs. distributed control, or the manner in which a new system is integrated into and interacts with existing systems.

An 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 (*APPLICATION DOMAIN* outcome, above). The strengths and weaknesses of a system option are generally not absolute, but vary with the application domain and other contextual factors.

- *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 should 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 realistic 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.

3.2.4 Systems Engineering Professionalism

Systems today are typically complex, requiring systems engineers to serve as productive and professional members of teams, using the most recent and appropriate technologies. There are three outcomes in this category that provide the necessary characteristics required for to be successful throughout their career: *PROFESSIONAL DEVELOPMENT*, *TEAMWORK*, and *ETHICS*.

- *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 both 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 to be able to analyze the advantages and disadvantages of leading edge concepts 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 it can teach the skills that enable lifelong learning.

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

  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 as well as internal and external stakeholders. 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 the SE process. 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.

- *ETHICS – Demonstrate knowledge of professional ethics and 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 be familiar with and be able to apply the relevant codes.

  An 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 an 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 3 shows the relationship between the recommended outcomes and sample objectives provided in Chapter 2; this is intended only as an example of how such an alignment exercise could be conducted during curriculum development. The alignment between outcomes and objectives is classified as either:

- Strong – There is clear correlation between achievement of an outcome and achievement of an objective; if the outcome is not reached by graduation 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.

Table 3. Outcome to Objective Relationships. (GRCSE Original)

	Objectives			
Outcomes	**1. SE Life Cycle**	**2. Multi-Disciplinary**	**3. Professionalism**	**4. Communication**
SE Concepts				
Foundation	Strong	Weak	Weak	Weak
Concentration	Strong	Weak	Weak	
Topic Depth	Strong	Weak	Weak	Weak
SE Role				
Application Domain	Strong	Weak	Weak	Strong
Specialty	Strong	Weak	Weak	Strong
Related Disciplines	Weak	Strong		Weak
Software in Systems	Weak	Strong		
SE Practice				
Requirement Reconciliation	Strong	Weak		Strong
Problem/Solution Evaluation	Strong	Weak		Strong
Realism	Strong	Strong	Weak	Weak
SE Professionalism				
Professional Development	Weak	Weak	Weak	Weak
Teamwork	Weak	Strong	Weak	Strong
Ethics			Strong	Weak

Note, a blank cell in Table 3 indicates no expected correlation between the outcome and the objective.

Chapter 2 provides a process for developing objectives and provides sample objectives, but not a set of recommended objectives because objectives should be program-specific. However, it is still useful for practitioners to understand the importance of alignment between outcomes and objectives. Table 3 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.

4.0 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 the knowledge, experience, and undergraduate degrees the GRCSE authors deem necessary for students entering master's programs in SE.

Attainment of program outcomes and objectives depends in part on student capabilities when they enter the program. For example, many topics in the CorBoK require a grasp of probability and statistics, such as risk management, safety engineering, and system verification and validation. A student entering a master's program with little or no understanding of probability and statistics is at a distinct disadvantage in classes which include these topics. Nevertheless, a program may for their own reasons opt to establish entrance requirements that are inconsistent with GRCSE recommendations. Like the other aspects of GRCSE, the entrance expectations are recommendations. Accordingly, a university may choose to implement a program with different entrance expectations than are recommended in GRCSE. If the entrance expectations have gaps with respect to the recommended entrance expectations, it may be necessary for the program to include additional subject matter in order to achieve the outcomes described in Chapter 3. If the actual entrance requirements of a program go beyond the recommended background described in GRCSE the additional capability of the students at entry will permit achievement of more ambitious outcomes than are described in GRCSE.

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. 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 entry (e.g., the Graduate Record Exam (GRE) in the US or the National Exam for Computing Graduation Entrance POSCOMP 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 U.S., 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[4], mathematics, or computer science.*

   GRCSE Preparatory Knowledge (Table 4, 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, other discipline engineers, and other professionals and be capable of understanding, supporting, and evaluating SE activities.

   Programs accepting students lacking the background in Table 4 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). Conversely, students with extensive experience in an engineering field may have acquired knowledge and capability in topics listed in Table 4.

   A student may also have earned a bachelor's (undergraduate) degree in SE. Accordingly, there are several possible approaches for students with an undergraduate degree in SE who wish to pursue a graduate degree in SE. 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, and
   - 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.).

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

Table 4. GRCSE Preparatory Knowledge. (GRCSE Original)

Knowledge Areas	Bloom Level*
Mathematics Fundamentals	
1. Probability and Statistics Basic probability theory, random variables and probability distributions, estimation theory, hypothesis testing, regression analysis, and analysis of variance.	Application
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).	Application
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.	Knowledge
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.	Application
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).	Application
General Education	
1. Oral and Written Communication Study and application of communication techniques that support development and enhancement of oral and written communication skills.	Application
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.	Comprehension

*The Bloom's Level definitions are discussed in Appendix C.

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 pertaining to 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 through experience.

   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.

   Two years is a low minimum requirement, and to fully appreciate systems engineering, a broad experience of five or more years is recommended for even better prepared students and to enable more fulfilling education for them.

   Qualifying SE functional areas of experience may include:

   - Requirements Engineering,
   - Risk and Opportunity Management,
   - Baseline Control,
   - Technical Planning,
   - Technical Effort Assessment,
   - Architecture/Design Development,
   - Qualification, Verification, and Validation,
   - Process Definition,
   - Tool Support,
   - Training,
   - Systems Integration,
   - Quality Assurance,
   - Specialty Engineering, and
   - Other.

   See (INCOSE 2011) and the SEBoK (BKCASE Editorial Board, 2015) for further descriptions of recognized SE activities.

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 very 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 aspects 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. The curriculum architecture, described in Chapter 5, Figure 4 provides a structure by which a university could address students who do not meet the entry expectations.

5.0 Curriculum Architecture

This chapter defines the curriculum architecture for GRCSE, which provides a basis to discuss and compare the architectures of actual SE programs at different universities with the GRCSE recommendations.

5.1 Influences on the GRCSE Curriculum Architecture

The curriculum architecture is predicated by the 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 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 KSAAs a student should learn in order to achieve the expected outcomes upon graduation with a master's degree 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 the following six elements:

- *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 his / her graduate education.
- *Foundation Knowledge* – The foundation is a set of KSAAs 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. This foundation is part of the *Core Body of Knowledge* (CorBoK) for GRCSE.

- *Concentration Knowledge* – Each student selects an 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 a specific program or particular educational institution focus and/or expertise.

- *Capstone Experience* – Each program expects students to demonstrate their accumulated KSAAs 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 capstone experience is an educational activity in which the student can both learn how to, and can demonstrate ability to, bind together all of the aspects of the program.

The foundation knowledge and the selected SE related concentration knowledge together form the CorBoK for GRCSE; the CorBoK is described in Chapter 6. Approximately 50% of the total time for an SE program is expected to be devoted to teaching the material in the CorBoK. The material in the CorBoK may be distributed in any of the courses comprising the program, and may be developed in more than 50% of the courses.

Figure 4 provides a visualization of the architecture. Time in Figure 4 notionally starts at the bottom and moves upward above the solid black baseline, which represents completion of the necessary preparatory work. That is, students would normally start with foundation topics and end with a capstone experience, though this is not always the case. The university is free to design courses that arrange teaching of material in any course blocks that the university finds appropriate. 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 4 is not strictly proportional to the amount of content for each GRCSE curriculum architecture element; however, the figure gives a sense of proportion of topic areas. The dark and mid-green shaded sections describe the entire content of the CorBoK. The capstone experience is intended to demonstrate the use of the knowledge gained from each of the topic areas. The intent of the capstone experience is primarily to implement and illustrate what a student has learned throughout the program. However, a student will certainly learn new things during the capstone experience. The capstone experience is really on a different dimension than the encompassed topics.

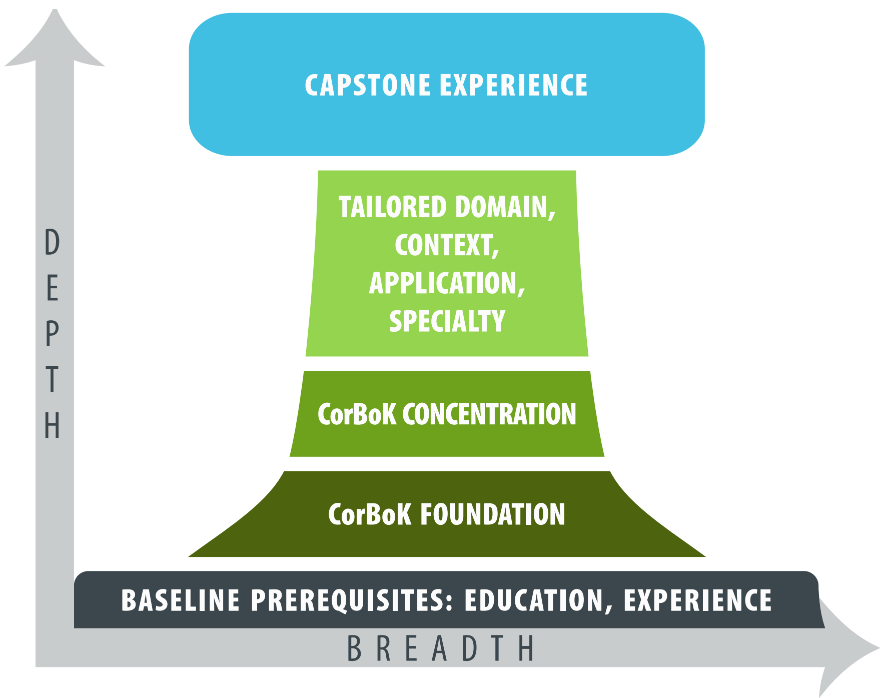


Figure 4. Architectural Structure of Topics in a GRCSE Master's Program. (GRCSE Original)

Building on Figure 4, Figure 5 gives an example showing how courses may relate to the architecture. The courses shown in Figure 5 are defined as follows:

- *Course 1* – Represented by three entries on the right side of the figure, covers foundation, concentration, and domain-specific or program-specific knowledge.
- *Course 2* – Integrates foundation and concentration knowledge.
- *Course3* – 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 on one topic area but can contain topics from several areas. There is no requirement 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.

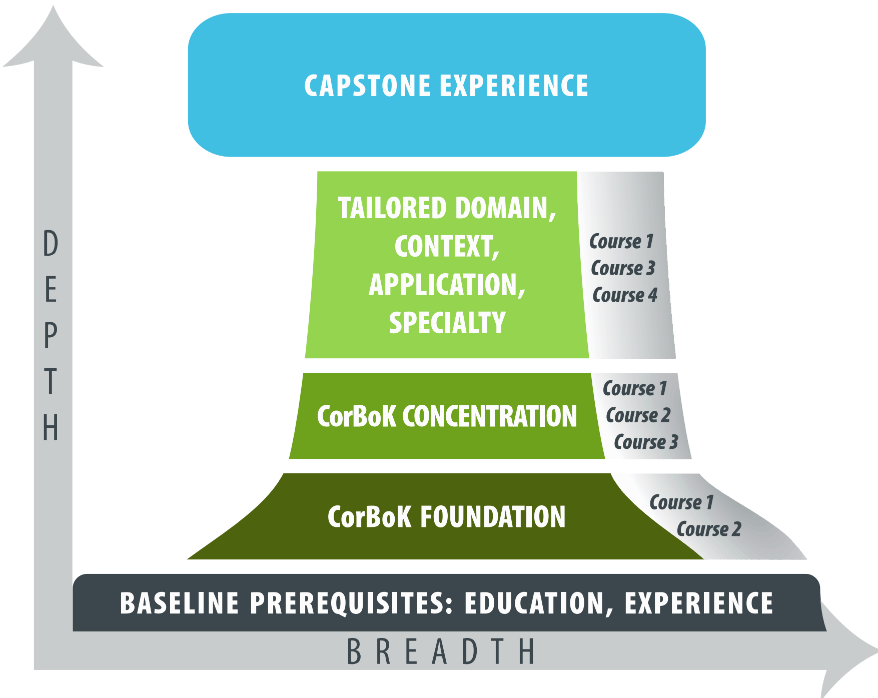


Figure 5. Course Alignment Which May Not Directly Correspond to Topics from Only One Area. (GRCSE Original)

6. The Core Body of Knowledge (CorBoK)

In a program that aligns with GRCSE, what specific SE knowledge are students expected to learn? GRCSE provides its answer in the form of a CorBoK, which is composed of knowledge that every graduate is expected to master. This chapter covers the topics that comprise the foundation and two concentration topics for the CorBoK. Appendix H includes the recommended minimum percentage of the curriculum anticipated in order to cover the CorBoK content at an appropriate level in a curriculum. The CorBoK does not directly address knowledge related to GRCSE's *APPLICATION DOMAIN* and *SPECIALTY* outcomes.

6.1 CorBoK Features

This section describes the major features of the CorBoK. These features should be viewed primarily as recommendations in order to implement a curriculum which supports the achievement of the outcomes specified in Chapter 3, especially in the cases of the *FOUNDATION*, *CONCENTRATION*, and *TOPIC DEPTH* outcomes.

6.1.1 CorBoK Foundation

The CorBoK is based on the Systems Engineering Body of Knowledge (SEBoK). The SEBoK, which acts 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 (KAs), 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 Introduction. Covers the scope, structure, uses, and evolution of the SEBoK.
- Part 2 – Foundations of Systems Engineering. Describes the characteristics of systems and foundation principles of SE.
- Part 3 – SE and Management. Addresses how SE is conducted and covers life cycle models and processes, SE development and evolution practices, management processes, and standards.
- Part 4 – Applications of SE. Covers the application of SE to the development and deployment of products, services, enterprises, and systems of systems.
- Part 5 – Topics on Enabling SE. 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. Focuses on the relationship of SE to other disciplines.
- Part 7 – SE Implementation Examples. Includes overviews of case studies and vignettes, which 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 from Parts 2 through 6 of the SEBoK and are listed in Tables 5 through 9 (in section 6.2). The Knowledge Areas and Topics in tables 5 through 9 are taken from SEBoK Version 1.4 (BKCASE Editorial Board, 2015). These tables provide a high-level description of the CorBoK; for additional detail the current SEBoK should be consulted. See Section 6.4 for further discussion of the impact on GRCSE of the continuing evolution of the SEBoK.

To help verify the significance and relevance of the SEBoK, topics from Parts 2-6 in the SEBoK were compared with information provided from the survey of graduate programs conducted at the beginning of the BKCASE project (see Appendix B). The survey shows that those programs with a "general systems focus" (66% of responding programs) included the following in their curricula: an overview of systems engineering concepts and principles (Part 2 of the SEBoK), and a more in-depth coverage of requirements, architecture, and management topics (Part 3 of the SEBoK). Topics in V&V, acquisition, deployment, and evolution were not evident in the survey, but most likely were covered as part of overview, management, or system definition courses. Based on this informal analysis, it appears that SEBoK Parts 2-6 contain topics suitable for the development of high-quality programs, as well as the evolution and advancement of existing programs.

It should be noted that it is expected that the GRCSE curriculum content will go beyond the CorBoK. Since there are few clear boundaries on systems, students need to be well-rounded across multiple disciplines; broad knowledge of different fields of study is highly valuable for a system engineer. The *DOMAIN*, *SPECIALTY*, and *RELATED DISCIPLINES* outcomes are particularly relevant to this issue.

6.1.2 50% Apportionment

Two principles have guided the GRCSE efforts: (1) to provide guidance that will support the creation and maintenance of effective system engineering programs, and (2) to allow for sufficient flexibility so that programs can pursue special interests and requirements. In support of these principles, it is recommended that the CorBoK comprise approximately 50% of the curriculum. The GRCSE authors feel a 50% apportionment allows a program to cover the CorBoK topics in sufficient breadth and depth, yet still provides the other 50% for a program to offer additional depth in selected CorBoK topics and to support program specialties and constraints. In particular, the CorBoK supports attainment of the *FOUNDATION*, *CONCENTRATION*, and *TOPIC DEPTH* outcomes.

Figure 6 depicts the organization of GRCSE curriculum material. Note again that 50% of the curriculum is devoted to the CorBoK Foundation and Concentration knowledge (described in Section 6.1.3), with the other 50% reserved for greater depth in the CorBoK topics and domain, application, or other knowledge that serves a program's or a student's special needs and interests. It should be emphasized that these knowledge areas do not correspond to courses, but represent the content that would be distributed across a program's courses. However, the Capstone Experience segment, depicted in Figure 6, would be delivered in one or more courses and represents the overarching nature of such a curriculum element, possibly covering many CorBoK areas and types of knowledge.

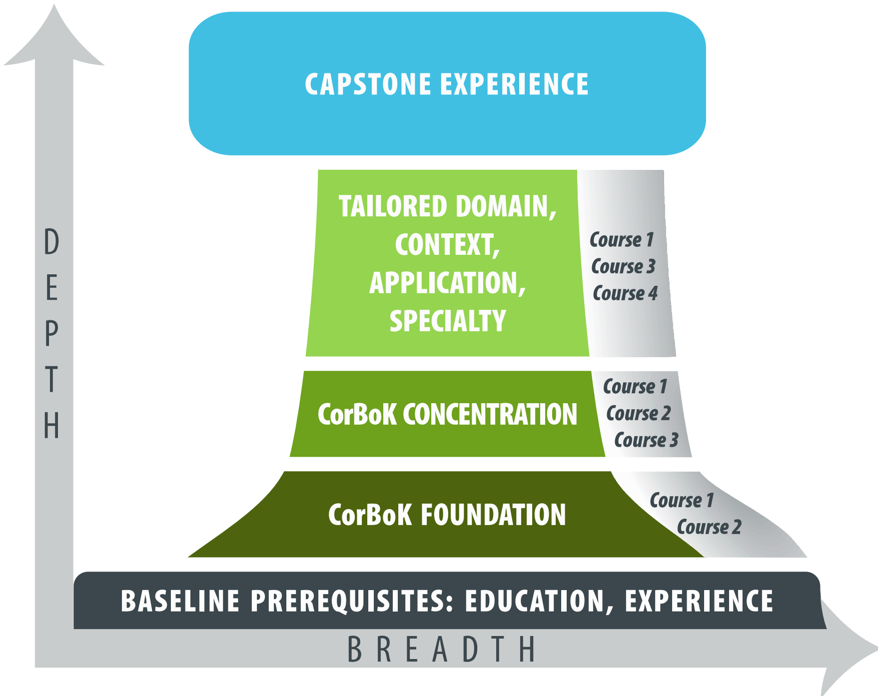


Figure 6. CorBoK Organization. (GRCSE Original)

6.2 CorBoK Organization

The CorBoK is structured in two parts: foundation knowledge and concentration knowledge. Foundation knowledge is composed of 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. The concentration knowledge builds on the foundation knowledge by revisiting foundation topics that should be learned in more depth, as well as by adding topics. GRCSE includes two concentrations: Systems Engineering Management (SEM) and Systems Design and Development (SDD). A program might choose from a wide array of concentrations; however, the following have been suggested: System Architecture, Verification & Validation, System Integration, System Evolution, In-Service SE, and Modeling & Simulation. The SEM and SDD concentrations are intended to illustrate this approach. Note that all SEM and SDD topics come from the SEBoK. Additional topics could be added for the concentrations as part of the 50% that goes beyond that specified in the CorBoK tables in this Chapter.

Tables 4 through 8 identify the foundation and concentration topics for SEBoK Parts 2 through 6 (BKCASE Editorial Board, 2015), recommending a Bloom's level of mastery for each, that together address the knowledge, skills, and abilities a student should possess to graduate with a master's degree in SE. In some cases, a Bloom's level is specified for a group of topics (e.g., all of the Systems Fundamentals topics in Table 4 are specified at the Comprehension level). If a concentration has a higher Bloom level than the foundation, that level is shown in the table; otherwise, the table cell is blank.

Appendix C explains Bloom levels in more detail, but fundamentally, they are a "depth of knowledge" metric: an ordered arrangement from "knowledge" (some acquaintance with the topic), to "comprehension" (an ability to discuss or explain a topic), to "application" (be able to use knowledge in the topic to solve a problem), to "analysis" (a deeper analytical understanding of the meaning of the topic, how it is organized and its importance). These levels are typically used by educators to set the level of attainment of educational or learning outcomes associated with an education unit or module. The CorBoK knowledge areas and topics (and the underlying SEBoK articles) support the learning outcomes since they represent the content in an SE curriculum, unit, or module.

The CorBoK Bloom level specification is a subjective judgment by educators and systems engineering professionals of the appropriate level to be achieved by all master's students. The GRCSE team arrived at the Bloom's levels through study, discussion, negotiation, specification, review, and revision. The Bloom specification should be viewed as general guidance, not as a precise or specific recommendation about how to structure and allocate the knowledge elements into a curriculum. Since the Bloom's recommendations apply to a KA, programs may have higher or lower Bloom levels for individual topics in the KA. Also, programs may offer greater depth as part of their tailoring.

The reader will notice significant difference in the Bloom's levels assigned to the topics across the various SEBoK parts. Topics in Parts 2, 5, and 6 are primarily listed at the Knowledge and Comprehension levels, while for Part 3 the Application or Analysis levels are used for many topics. The reason for this is related to the nature of the parts:

- Part 2 topics are primarily conceptual, with the concepts supporting the topics in Part 3.
- Part 3 concentrates on the processes, methods, and practices used to manage, develop, operate and maintain systems.
- Parts 4, 5 and 6 contain topics that explore special categories of systems engineering, important engineering support and enabling features, and background on related disciplines. One exception is the Enabling Teams topic in Part 5; it is felt that all systems engineers need Application capability for this topic.

Table 5. CorBoK Bloom's Levels for Part 2: Foundations of SE Topics. (GRCSE Original)

Knowledge Area	Part 2 Topic	Foundation	SEM	SDD
Systems Fundamentals	What is a System?	Comprehension		
	Types of Systems			
	Groupings of Systems			
	Complexity			
	Emergence			
Systems Science	History of Systems Science	Knowledge		
	Systems Approaches			
Systems Thinking	What is Systems Thinking?	Knowledge		
	Concepts of Systems Thinking			
	Principles of Systems Thinking			
	Patterns of Systems Thinking			
Representing Systems with Models	What is a Model?	Knowledge		
	Why Model?			
	Types of Models			
	System Modeling Concepts			Application
	Integrating Supporting Aspects into System Models			
	Modeling Standards			
Systems Approach Applied to Engineering	Overview of the Systems Approach	Knowledge*		
	Engineered System Context			
	Identifying & Understanding Problems & Opportunities			
	Synthesizing Possible Solutions			
	Analysis and Selection between Alternative Solutions			
	Implementing and Proving a Solution			
	Deploying, Using, and Sustaining Systems to Solve Problems			
	Stakeholder Responsibility			
	Applying the Systems Approach			

*The topics in this KA largely have counterparts in Part 3 for which recommended mastery is at the higher Bloom levels of Comprehension, Application, and Analysis. Table 5 captures these recommendations.

Table 6. CorBoK Bloom's Levels for Part 3: SE and Management. (GRCSE Original)

Knowledge Area	Part 3 Topic	Foundation	SEM	SDD
Introduction to Life Cycle Processes	Generic Life Cycle Model	Application		
	Applying Life Cycle Processes			
	Life Cycle Processes and Enterprise Needs			
Life Cycle Models		Application		
	System Life Cycle Process Drivers and Choices			
	System Life Cycle Process Models: Vee			
	System Life Cycle Process Models: Iterative			
	Integration of Process and Product Models			
	Lean Engineering			
Concept Definition	Business or Mission Analysis	Application		Analysis
	Stakeholder Needs and Requirements			
System Definition	System Requirements	Application		Analysis
	System Architecture			
	Logical Architecture Model Development			
	Physical Architecture Model Development			
	Systems Design			
	System Analysis			
System Realization	System Implementation	Application		Analysis
	System Integration			
	System Verification			
	System Validation			
System Deployment and Use	System Deployment	Comprehension		Application
	Operation of the System			
	System Maintenance			
	Logistics			
SE Management	Planning	Comprehension	Analysis	
	Assessment and Control			
	Risk Management			
	Measurement			
	Decision Management			
	Configuration Management			
	Information Management			
	Quality Management			
Product and	Service Life Extension	Comprehension	Analysis	Application

Knowledge Area	Part 3 Topic	Foundation	SEM	SDD
Service Life Management	Capability Updates, Upgrades, and Modernization			
	Disposal and Retirement			
SE Standards	Relevant Standards	Comprehension		
	Alignment and Comparison of the Standards			
	Application of SE Standards			

Table 7. CorBoK Bloom's Levels for Part 4: Applications of SE. (GRCSE Original)

Knowledge Area	Part 4 Topic	Foundation	SEM	SDD
Product SE	Product SE Background	Knowledge*		
	Product as a System Fundamentals			
	Business Activities Related to Product SE			
	Product SE Key Aspects			
	Product SE Special Activities			
Service SE	Service Systems Background	Knowledge*		
	Fundamentals of Services			
	Properties of Services			
	Scope of Service SE			
	Value of Service SE			
	Service SE Stages			
Enterprise Systems Engineering	Enterprise SE Background	Knowledge*		
	The Enterprise as a System			
	Related Business Activities			
	Enterprise SE Key Concepts			
	Enterprise SE Process Activities			
	Enterprise Capability Management			
Systems of Systems (SoS)	Architecting Approaches for Systems of Systems	Knowledge*		
	Socio-Technical Features of Systems of Systems			
	Capability Engineering			

*It is recommended that one of these four knowledge areas be extended at least one level higher. This may be achieved in the University specialization aspect of the program.

Table 8. CorBoK Bloom's Levels for Part 5: Enabling SE. (GRCSE Original)

<table>
<tr><th>Knowledge Area</th><th>Part 5 Topic</th><th>Foundation</th><th>SEM</th><th>SDD</th></tr>
<tr><td rowspan="6">Enabling Businesses and Enterprises</td><td>Systems Engineering Organizational Strategy</td><td rowspan="6">Knowledge</td><td rowspan="6">Comprehension</td><td rowspan="6"></td></tr>
<tr><td>Determining Needed SE Capabilities in Business & Enterprises</td></tr>
<tr><td>Organizing Business and Enterprises to Perform SE</td></tr>
<tr><td>Assessing SE Performance of Business & Enterprises</td></tr>
<tr><td>Developing SE Capabilities within Businesses & Enterprises</td></tr>
<tr><td>Culture</td></tr>
<tr><td rowspan="3">Enabling Teams</td><td>Team Capability</td><td rowspan="3">Application</td><td rowspan="3">Analysis</td><td rowspan="3"></td></tr>
<tr><td>Team Dynamics</td></tr>
<tr><td>Technical Leadership in Systems Engineering</td></tr>
<tr><td rowspan="4">Enabling Individuals</td><td>Roles and Competencies</td><td rowspan="4">Comprehension</td><td rowspan="4">Application</td><td rowspan="4"></td></tr>
<tr><td>Assessing Individuals</td></tr>
<tr><td>Developing Individuals</td></tr>
<tr><td>Ethical Behavior</td></tr>
</table>

Table 9. CorBoK Bloom's Levels for Part 6: Related Disciplines. (GRCSE Original)

<table>
<tr><th>Knowledge Area</th><th>Part 6 Topic</th><th>Foundation</th><th>SEM</th><th>SDD</th></tr>
<tr><td rowspan="4">SE and Software Engineering (SwE)</td><td>The Nature of Software</td><td rowspan="4">Comprehension</td><td rowspan="3"></td><td rowspan="4">Application</td></tr>
<tr><td>An Overview of the SWEBOK Guide</td></tr>
<tr><td>Ten Things a Systems Engineer Needs to Know about Software Engineering</td></tr>
<tr><td>Ten Things a Systems Engineer Needs to Know about Managing a Software Team</td><td>Application</td></tr>
<tr><td rowspan="4">SE and Project Management (PM)</td><td>The Nature of PM</td><td rowspan="4">Comprehension</td><td rowspan="4">Application</td><td rowspan="4"></td></tr>
<tr><td>Overview of PMBOK® Guide</td></tr>
<tr><td>Relationships between SE & PM</td></tr>
<tr><td>The Influence of Project Structure and Governance on SE and PM Relationships</td></tr>
<tr><td>SE and Industrial Engineering</td><td>SE and Industrial Engineering</td><td>Knowledge</td><td></td><td></td></tr>
<tr><td>SE and Procurement/ Acquisition</td><td>SE and Procurement/Acquisition</td><td>Knowledge</td><td></td><td></td></tr>
<tr><td rowspan="2">SE and Specialty</td><td>Integration of Specialty Engineering</td><td rowspan="2">Comprehension</td><td>*</td><td>*</td></tr>
<tr><td>Reliability, Availability, and</td><td>*</td><td>*</td></tr>
</table>

Knowledge Area	Part 6 Topic	Foundation	SEM	SDD
Engineering	Maintainability			
	Human System Integration		*	*
	Safety Engineering		*	*
	Security Engineering		*	*
	System Assurance	Knowledge	*	*
	Electromagnetic Interference/Electromagnetic Compatibility		*	*
	Resilience Engineering		*	*
	Manufacturability and Producibility		*	*
	Affordability		*	*
	Environmental Engineering		*	*

*GRCSE recommends that programs choose one or more Specialty Engineering KAs and teach them at the Application level or higher. These KAs may represent a special focus of a program or individual student interest. This level may be achieved in the apportionment of the curriculum that extends beyond the 50% devoted to the CorBoK and may support the *SPECIALTY* outcome.

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. Substantial instructional activities may not be needed to achieve these levels.
- There are 20 topics at the Application level in the Foundation and 3 topics at the Analysis level. To achieve these levels, student activities related to problem solving and inquiry will have to be included in the curriculum.
- SEM elevates 11 topics from the Comprehension level to the Application level and elevates 11 to the Analysis level; hence, SEM requires 26 topics be covered at the Application level and 11 at the Analysis level.
- SDD elevates 11 topics from the Comprehension level up to the Application level and elevates 10 topics from the Application level to the Analysis level; subsequently, SDD requires 31 topics to be covered at the Application level and 13 at the Analysis level.
- The SEBoK includes recommended primary and additional readings for every topic. These recommendations are natural sources for textbooks and papers to be used in courses that cover the CorBoK topics.

6.3 Recommended Distribution of CorBoK 50%

Based on the survey of graduate programs conducted at the beginning of the BKCASE project (see Appendix B), the SE programs requires the equivalent of ten to twelve 3-unit semester courses, which expands to approximately 450 to 500 contact hours between students and faculty in whatever format the courses are delivered. For purposes of planning curriculum content, GRCSE estimates 480 hours as the total number of contact hours needed for a SE master's program. Consequently, using the 50% guideline, CorBoK instruction is presumed to take approximately 240 contact hours, including examinations and non-instructional time spent with students. Clearly, this will vary from program to program, but the general guidance of 240 contact hours for coverage of CorBoK topics provides a key guidance element in GRCSE curriculum design.

There is the question: How should these 240 contact hours be distributed across the curriculum? That is, how should the hours be allocated to the various Parts, KAs, and Topics? Considering the variation in the Bloom levels specified for topics in Parts 2-6, it is clear that Part 3 will require more instruction time than do Parts 2, 4, 5, and 6. The GRCSE authors engaged in much analysis and discussion about what guidance should be provided on how the 240 hours should be distributed across the CorBoK topics. Although no conclusion on precise guidance was reached, the GRCSE authors recommend that the CorBoK's 50% content be allocated as follows: approximately 30 % of the time should be devoted to Part 3 CorBoK topics and the remaining 20% should be devoted to topics in Parts 2, 4, 5, and 6. See Appendix H for an example of how the 240 hours might be distributed.

6.4 Development of the CorBoK

The Knowledge Areas and Topics in tables 5 through 9 are taken from SEBoK Version 1.4 (BKCASE Editorial Board, 2015). The SEBoK is under continual review and updated versions are produced twice per year. The topic level mappings will remain valid as the detailed material evolves to match the changing nature of SE knowledge. Anyone using GRCSE as a curriculum reference will need to track these evolutions of SEBoK content and decide how to respond to them.

If SEBoK topics are added or renamed, or if content changes affect the CoRBoK knowledge levels, a new version of GRCSE will be produced to represent these changes. Due to difference in publishing timelines this may not be done immediately and GRCSE users need to be aware of the latest SEBoK topics. Each new version of the SEBoK includes a detailed description of what has changed. Notification of new versions is provided by the SEBoK Newsletter, which is available via www.BKCASE.org.

The GRCSE authors went through several iterations to produce the recommendations shown in this chapter, collecting inputs from numerous sources and iterating in author workshops. To this end, the GRCSE team, over a series of meetings determined the Bloom level specifications and estimated contact time needed to cover different CorBoK parts and their KAs.

The process was collaborative in nature and the methodology involved a quasi-Wideband Delphi technique. Reviewer feedback has been helpful in evolving the recommendations into more mature model. Anyone wishing to provide additional feedback on the value of the CoRBoK should contact the BKCASE editorial board at www.BKCASE.org.

7. Implementation

GRCSE consists of the process for developing objectives, outcomes, entrance expectations, architecture, and the CorBoK. This chapter discusses how an institution may consider incorporating 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.

7.1 Understanding Initial Conditions

Each institution will begin 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 few or none. The quality of the existing elements may or may not be satisfactory. 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 constituents 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, programs are unlikely to develop 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 et al. (2011)[5] provides a useful table for assessing program maturity with respect to GRCSE. It is reprinted below as Table 10. Programs are categorized as initial, emerging, developed, or highly developed.

Table 10. . Systems Engineering Program Development Phases Mapped to GRCSE (Squires et al. 2011). Released by the American Society for Engineering Education (ASEE).

	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.

[5] From “Developing Systems Engineering Graduate Programs Aligned to the Body of Knowledge and Curriculum to Advance Systems Engineering (BKCASE™) Guidelines,” which was presented at the American Society for Engineering Education (ASEE).

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

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

7.2 Developing Objectives

Program objectives should be developed or updated through consultation with stakeholders. It is critical that they remain consistent with the mission of the institution, provide information published in a form accessible to the public, articulate the goals of the program in a form that is testable, and provide information that is 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; therefore, they do not formally develop objectives, even though the departments responsible may maintain them for their undergraduate programs. GRCSE specifically recommends the discipline of developing and assessing program objectives for SE graduate programs. Most departments 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.

7.3 Developing Outcomes

Many GRCSE expected outcomes explicitly require tailoring to a particular program. For example, the *CONCENTRATION* outcome requires the program to select a concentration area. The *TOPIC DEPTH*, *APPLICATION DOMAIN*, and *SPECIALTY* outcomes 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 curriculum committees are the usual mechanism for tailoring and designing these outcomes.

7.4 Addressing the CorBoK

The CorBoK lists over a hundred topics extracted from the SEBoK and recommends a minimum Bloom's level for each topic, both for the foundation level and the concentrations. It is expected that about half the student contact hours in the curriculum would be required to attain these levels, leaving the other half for program unique content (see Figure 6 in Chapter 6). Much of the local specialization involves utilizing topics in the CorBoK to achieve a higher attainment than the minimum. Additionally, the CorBoK tables can be used as a framework for planning that specialization. For example, the Specialty outcome could be addressed by the selection of one or two specialty domains of focus from the CorBoK.

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

7.5 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 may also reduce the number of students eligible for a program. Relaxing entrance requirements, on the other hand, may result in either increased preparation work (increasing program length), or reduced levels of attainment of the outcomes. Each program must consider its institutional mission, stakeholder needs, accreditation requirements, and market forces when setting its particular entrance requirements.

In particular, the work experience entrance expectation should be carefully considered. A lack of engineering experience affects the team outcomes and much of the expected background for learning the CorBoK. Programs that relax the entrance expectations for some but not all students may seek to develop a set of preparatory courses to address relevant areas.

The domain specialization selected for the *APPLICATION DOMAIN* outcome may require additional technical prerequisites for admission.

7.6 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; programs courses and the capstone experience should be focused on closing the gaps between students' expected capabilities upon entrance and upon graduation.

Some design considerations include the use of design problems that span multiple courses versus small problems within individual courses, the use and tailoring of existing courses versus the development of new ones, and the length and type of capstone experience.

The CorBoK is based on the SEBoK. Once decisions have been made about the specific topics to be covered in a program, program developers can refer to the SEBoK for additional design support. For each topic and knowledge area (KA), the SEBoK includes primary and additional references. Those references provide obvious candidates for textbooks, journal articles, and conference papers which can be used in individual courses. For additional information on how faculty may use the SEBoK to support curriculum design, please see SEBoK Use Case 4: Educators and Researchers in Appendix G.

7.7 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, and the identification of individuals responsible for providing evidence, and process management.

Elements assessed include those items on the right hand of the Vee in Figure 3 in Chapter 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. Likewise, 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.

7.8 Continuous Improvement

Assessments are wasted effort 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 illustrating the success of earlier efforts. An example 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). Assessments of programs should be timed to enable the fastest possible response to findings given the university program approval cycle.

7.9 Examples of Implementation

In future, the implementation experiences of early adopters of GRCSE will be documented and lessons learned captured.[6] Detailed examples of the curricular design, implementation, costs, and assessment should be included in these examples.

To see community discussion on implementation experiences, please see www.bkcase.org/grcse/.

To share a GRCSE implementation experience, contact the authors at bkcase@stevens.edu.

[6] These will be maintained separately from the body of GRCSE, but will be available to the community. The exact mechanism for compiling these documents has not yet been determined.

8. Assessment

Assessment is essential for universities to determine how well their programs achieve their outcomes and objectives. It is an important part of a continuous improvement plan used to improve the quality of an educational program. This chapter introduces major methods and principles that may be used as support and to provide guidance to program leaders in performing appropriate assessment and evaluation activities. Additional detail is provided in Appendix E.

8.1 Program Assessment

How does a program evaluate whether its educational objectives and desired outcomes are being achieved? It employs a 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.
- Determine how well student activities, products, and performance provide evidence that the student has met outcomes and objectives.
- Determine if curriculum and instruction are facilitating outcome achievement by students.
- Provide accreditation authorities (including the US Accreditation Board for Engineering and Technology (ABET), French Commission des Titres d'Ingenieur (CTI), United Kingdom (U.K.) Engineering Council, the German Accreditation Agency for Degree Programs in Engineering, Informatics/Computer Science, the Natural Sciences, and Mathematics (ASIIN), and the German Accreditation, Certification and Quality Assurance Institute (ACQUIN), among others) with information these authorities deem mandatory.
- Take a central role in global processes for improving program quality.

GRCSE addresses assessment in two complementary ways:

- In Appendix E, GRCSE presents assessment as an iterative four-step management method which is 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 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 also to 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.

Achievement of program outcomes is demonstrated using assessments of course outcomes; achievement of course outcomes is demonstrated using assessments of individual student performance. So the "targets" of assessment cover a broad scope and should not be considered in isolation. For more detail, see Appendix E (E.1).

The following sections show how assessment is developed by some representative schools and universities that teach SE. They cover assessment of program objectives, program outcomes, courses, and students. It begins with a discussion of the assessment of program objectives, followed by program outcomes, then courses, and finally concludes with student assessment.

8.2 Assessing Program Objectives

Program objectives concern what graduates are expected to attain within a few years of graduation. Chapter 2 described a process for setting program objectives accompanied by a generic set of examples.

Measuring the attainment of objectives directly is difficult because graduate work products are not usually 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 pertaining to the graduate (e.g., year of graduation and program studied) and the graduate's performance; they may also take this information to compare with program objectives. Beyond individual assessment, performance indicators could cover the assessment of graduates in the context of the company (for example, a graduate's length of time to reach job productivity within their new systems engineering role, additional training investment consideration to achieve graduate readiness, etc.). Locating and getting good response rates from graduates and their employers can be challenging; although the use of institutional alumni organizations can be helpful. Another challenging point can concern the selection of representative companies in order to reduce volume of data that must be handled.

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. However, care must be taken to address the potential impact of the sampling bias in having the same participants in the creative and assessment activities.

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

To begin, Sim U asks graduates within 5 years of graduation to indicate if the following statements are true: “I have clearly reached the following Sim U SE objective [insert a program objective]”. Examples may be as follows:

- *I currently employ SE methods taught by Sim U: requirement management, architecture definition, etc.*

  True or False

- *I currently hold a leadership position of technical director or division chief or chief engineer or system program manager.*

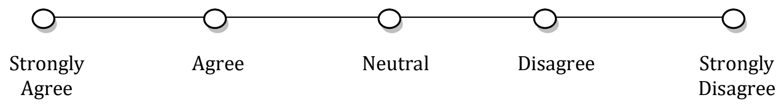
  True or False

Next, Sim U asks graduates to indicate their agreement or disagreement, on an ordinal Likert (1932) scale[7], with statements such as “The Sim U Systems Engineering program provided me with an educational foundation that allowed me to [insert a program objective]”. Examples may be as follows:

- *The Sim U Systems Engineering program provided me with an educational foundation that allowed me to take a leadership role in system engineering department of my company.*

  Strongly Agree –1—2—3—4—5—6—7 Strongly Disagree

- *The Sim U Systems Engineering program provided me with an educational foundation that prepares me for assignment related to re-engineering of legacy systems.*



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

The category 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. With large enough numbers of graduates it would be possible to survey different groups in a cycle, reducing fatigue effects whilst enabling long horizon studies of sample progression.

7 The Likert scale, named after the psychologist Rensis Likert, provides an ordinal scale (1-3, 1-5, 1-7, etc.) to measure strength of attitudes.

8.3 Assessing Program Outcomes

Chapter 3 described the GRCSE program outcomes. Program outcome assessment focuses on the competencies and capabilities of students at graduation to determine if the program is meeting its outcomes. During this process, potential improvements to the curriculum and instruction are identified, implemented, and their effectiveness tracked.

<u>Step 1: Identify Program Outcomes</u>

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; in addition, other programs misclassified objectives or courses taken as outcomes, or stated that outcomes were under development.

Continuing the example, Sim U develops the following outcome, based on the *TOPIC DEPTH* outcome, as one of the outcomes for its SE program:

Step 1.1: Achieve Bloom's Synthesis Level of Attainment for System Analysis.

Outcomes are 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 typically 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. The order of the scales can be defined with "1" as either the best or the worst score, depending on preference.

<u>Step 2: Define Assessment Strategy and Methods</u>

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.

For each of the four performance measures above, a rubric is developed that provides a word description for each numeric score. For example, consider the performance measure "Develop measures for a trade-off study" and an associated five point Likert Scale:

Table 11. Example Rubric for Performance Measure for Outcome Assessment. This example uses a five-point Likert scale, with lower numbers being better. (GRCSE Original)

Description	Likert score
All key dimensions of the trade-space are described with valid, quantifiable measures.	1
The set of measures identified describes most of the key system attributes. Measures are well-written and quantifiable.	2
A few measures are identified, and some are descriptive of the key systems attributes. The measures are quantifiable.	3
At least a single measure identified, but the measure is not descriptive of the required system attribute. The measure may not be quantifiable.	4
No measure identified.	5

Sim U chooses not to involve employers of graduates, although it would have given an interesting external point-of-view administered through employer surveys or by 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 samples from the population of student work for that quarter. It also decides to assess every student thesis or capstone project throughout the year.

Step 2.3: Define the Organization for an Assessment

Sim U establishes a faculty committee to perform the assessments, reducing instructor self-assessment bias and increasing consistency. 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 assessments indicate that the program is meeting the outcome target for each performance measure. Accordingly, Sim U decides that it does not need to take 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. (GRCSE Original)

Student Outcome: *Achieve a Bloom's synthesis level of attainment for system analysis. (This example uses a five-point Likert scale, with higher numbers being better.)*

Performance Criteria	Courses with Coverage	Method(s) of Assessment	Where Data are Collected	Length of Assessment Schedule	Year / Semester of Data Collected	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 Assessing Course Delivery

Courses have learning goals and teaching methods. Assessing a course brings these to light and allows colleagues to collectively examine practices that otherwise may have remained 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, especially within the air transport sector.

Step 1: Identify Course Outcomes

The school defines expected outcomes for each course in the syllabus. The outcomes use 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.

Step 2: Define the 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 this method of assessing is provided for mapping course objectives to course assessment methods in Table 13. How the school performs student assessment is covered in the next section.

Table 13. Possible Methods for Assessing Bloom's Levels of Achievement for Outcomes. (GRCSE Original)

Course Reference	Course Outcomes	Student Assessment Method
SE3: Introduction to V&V	Upon successfully completing the course, students will be able to describe and explain basics definitions and objectives of verification and validation processes	Multiple choice examination questions
SE3: Introduction to V&V	Upon successfully completing the course, students will know the four basic means for supporting verification and validation processes: review, demonstration, analysis, and test	Short answer examination questions
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 • the definition of the organization	Course project: Develop a V&V plan
SE3-1: Introduction to Formal Methods	Upon successfully completing the course, students will have an in-depth knowledge of the model based system engineering approach, including: • model checking approach	Course project: MBSE

Course Reference	Course Outcomes	Student Assessment Method
	• proof model approach	
SE3-2: Introduction to Testing	Upon successfully completing the course, students should have an in-depth knowledge in test techniques, including: • white-box and black box coverage • test tools and environment	Essay examination question

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 14.

Table 14. Example of a Teacher Online Survey. (GRCSE Original)

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 computed for low levels of satisfaction (1/4 and 2/4) and high levels of satisfaction (3/4 and 4/4). Each topic in which the number of low levels is greater than the number of high levels is investigated and in most cases, corrective actions are put in place. Moreover, when this kind of survey is repeated over the years on the same course, it becomes possible to follow trends in quality indicators, as well as to review the effectiveness of corrective actions. Table 15 below shows the kind of results obtained.

Table 15. Example of Student Online Survey Results – Extract. (GRCSE Original)

Course: xxxx	Teacher: xxxx # of Forms: xxxx			
Topic	**1/4**	**2/4**	**3/4**	**4/4**
Course				
Knowledge of the course objectives	1	13	58	32
Suitability course/objectives	2	12	56	34
Ease of assimilation	4	30	40	30
Course plan	1	9	42	52
[additional questions]				
Project				
Tutor availability	18	15	39	32
Difficulty of the subject	6	6	27	65
Attractiveness of the subject	8	8	31	57
[additional questions]				
Test				
Exam level	0	12	46	46
Correlation with lessons	0	2	46	56
[additional questions]				
Appraisal				
General impression	2	16	48	38
Personal work	12	48	36	8
[additional questions]				

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
- course evaluation, as the information about student achievement helps in determining the level of attainment of course outcomes

The aeronautical program example, from the previous section "Assessing Course Delivery", 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. Next, the black-box exercise is described. Its objective 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's "Knowledge" level. Each student has 40 minutes to answer 40 questions.
- Problem-solving for Bloom's "Comprehension" level. In a written test, each student is asked to analyze a case study as well as to propose a solution. This problem-solving examination lasts two hours.
- A capstone project for Bloom "Application" level. The project is conducted by a team whose oral presentation, product creation, and reports are assessed.

  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 receive an individual assessment, in which they are required to perform a part of the presentation and to answer questions about another student's part of the presentation.

Table 16 is a sample matrix for assessing Bloom's levels of achievement against program outcomes, course outcomes, and the assessment methods for performing the measurement.

Table 16. Possible Methods for Assessing Bloom's Level of Achievement. (GRCSE Original)

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 describe and explain 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	Application	Capstone (team) project, including: - Oral presentation - Product creation - Written reports

Program Outcomes	Course Outcomes	Bloom Level	Student Assessment Method
	life cycle • the specification of the environment • 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 • 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 • test tools and environment	Comprehension	- Problem-solving

9. Evolution

GRCSE v. 1.1 incorporates a set of small changes to GRCSE v. 1.0. The changes in the fore matter up to and including the Executive Summary more correctly present the current arrangements for support and maintenance with respect to the Board of Governors, the sponsoring organizations and the BKCASE Editorial Board. Changes to the main text of GRCSE are responses to changes in SEBoK up to v. 1.4 and corrections of minor textual matters found since the v. 1.0 release.

GRCSE v. 1.0 was the first release offered for world-wide use and adoption. It is aligned with SEBoK v. 1.0, which was released in September 2012, and is built on GRCSE v. 0.5, which was released for world-wide comment in December 2011. The BKCASE project, originally led by the Stevens Institute of Technology and the Naval Postgraduate School, was responsible for the development of both GRCSE and the SEBoK.

The BKCASE products are now under the long-term stewardship of a Board of Governors composed of the International Council on Systems Engineering (INCOSE), the IEEE-Computer Society (IEEE-CS), and the Systems Engineering Research Council (SERC) for joint promulgation, maintenance, and evolution. Both the SEBoK and GRCSE, if they are to have widespread impact, need the influence and long-term stable resources that only major professional societies can offer. INCOSE (the most prominent global professional society dedicated to SE) and the IEEE-CS (the computing related element of the largest and most prominent global professional engineering society) are ideal organizations to be co-stewards of the SEBoK and GRCSE projects. As the third steward, the SERC provides continuity from the development phase of both SEBoK and GRCSE and includes many universities with systems engineering educational programs.

INCOSE's interest in becoming a co-steward with IEEE-CS has been strong since the project began. The majority of the original BKCASE authors were INCOSE members, many in leadership positions. The INCOSE Board of Directors has repeatedly shown high interest in the BKCASE project. During its initial development, INCOSE funded three authors to contribute to the writing as well as to represent INCOSE. The Board of Directors has passed multiple resolutions expressing INCOSE's commitment to be a steward of both the SEBoK and GRCSE and to authorize the funds necessary for that stewardship to be successful.

From the beginning of the project, the IEEE also showed strong interest in becoming a steward; however, because the IEEE is much larger than INCOSE, they needed to identify the specific organizational unit to work with SERC and INCOSE. The IEEE-CS is that unit. The leadership of the IEEE-CS had representatives on the original 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 the Stevens Institute of Technology (representing the SERC), INCOSE, and the IEEE-CS. That agreement was signed by all parties in Spring 2013. An amended version was produced in June 2014; this clarified a few things, but did not fundamentally change the agreement or extend the period of the agreement. A new version of the MoU, extending the agreement to 2016-2018, will be signed by the end of 2015.

The agreement covers how the products are disseminated and supported, the periodicity for minor and major revisions, as well as who can authorize those revision. Additionally, it confirms that all product revisions are free for worldwide use.

The major points of agreement are that to promote stability, GRCSE will be infrequently updated. The current, minor-update, is the first update to GRCSE. The next update is planned for the end of 2016 or early 2017. In the meantime, the stewards will collect any comments on the BKCASE site using a DISQUS comment tool. The URL for making comments is: www.bkcase.org/grcse/.

Editorial control of the SEBoK and GRCSE is now held by an editorial board with an editor-in-chief appointed by the stewards. One of the primary functions of the editorial board is to manage the evolution of GRCSE. In particular, the editorial board reviews and adjudicates all user feedback, and manages revisions to ensure that GRCSE increases in value over time both to the stewards and to the larger SE community.

Glossary

List of Acronyms

Acronym	Definition
ACM	Association for Computing Machinery.
ASIIN	Accreditation Agency for Degree Programs in Engineering, Informatics/Computer Science, the Natural Sciences, and Mathematics (Germany)
BKCASE	Body of Knowledge and Curriculum to Advance Systems Engineering
CNES	Centre National d’Etudes Spatiales Astrium (EADS)
CorBoK	Core Body of Knowledge
CSEP	Certified SE Professional (used by INCOSE)
CSER	Conference on Systems Engineering Research
DGAC	Direction Generale de l’Aviation Civile (France)
DoD	U.S. Department of Defense
EADS	European Aeronautical Defence and Space
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
KSAA	Knowledge, Skills, Abilities, and Attributes
OSD	Office of the Secretary of Defense (DoD)
NDIA	U.S. National Defense Industrial Association
PMBOK®	Project Management Body of Knowledge®
POSCOMP	National Exam for Computing Graduation Entrance (Brazil)
SDD	System Development or Design
SE	Systems Engineering

Acronym	Definition
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 cognitive 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— An observable, measurable set of skills, knowledge, abilities, behaviors, and other characteristics an individual needs to successfully perform work roles or occupational functions. (OPM 2012) (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. (Pyster and Olwell et al. 2012)

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 KSAAs 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 factors—The systematic application of relevant information about human abilities, characteristics, behavior, motivation, and performance. It includes principles and applications in the areas of human related engineering, anthrometics, ergonomics, job performance skills and aids, and human performance evaluation. (INCOSE 2011, 363)

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— Acquaintance with facts, truths, or principles, as from study or investigation. (Dictionary.com. 2012)

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 organizational element is parts) and consists of one or more topics and subtopics.

knowledge, skills, abilities, and attitudes (KSAA) –What 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. (Pyster 2009, 73); (2) The evolution of a system, product, service, project or other human-made entity from conception through retirement. (ISO/IEC 2008); (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[8] 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

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

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 set of elements in interaction.* (von Bertalanffy 1968); *(2) A combination of interacting* **system elements** *organized to achieve one or more stated purposes.* (ISO/IEC/IEEE 2008)

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)— *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.* (ISO/IEC/IEEE 2010); *(2) 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*
- *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 2012)

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).

References

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

ABET Engineering Accreditation Commission. 2010. "Criteria for accrediting engineering programs, 2011-2012." Accreditation Board for Engineering and Technology (ABET) website. 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.

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.

BKCASE Editorial Board. 2015. *The Guide to the Systems Engineering Body of Knowledge (SEBoK)*, v. 1.4. R.D. Adcock (EIC). Hoboken, NJ: The Trustees of the Stevens Institute of Technology. Accessed October 25 2015. www.sebokwiki.org. BKCASE is managed and maintained by the Stevens Institute of Technology Systems Engineering Research Center, the International Council on Systems Engineering, and the Institute of Electrical and Electronics Engineers Computer Society.

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 D.R. Krathwohl. 1956. *Taxonomy of educational objectives the classification of educational goals handbook I: Cognitive domai*n. London, UK: Longman Group Ltd.

Brainerd, J.G. 1962. Editorial. *IRE Transactions on Education* 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.

Chang, C.M. 2010. *Service Systems Management and Engineering: Creating Strategic Differentiation and Operational Excellence*. New York, NY, 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.

Charette, R.N. 2005. "Why software fails." *IEEE Spectrum*. September 2005. Accessed November 17, 2011. http://spectrum.ieee.org/computing/software/why-software-fails

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.

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

Dictionary.com. 2012 "Knowledge." Available at http://dictionary.reference.com/browse/knowledge.

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.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. “Section 2.” In *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. and 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).

Howard, R.A. 1962. Mathematical programming in systems engineering. *IEEE 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.

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

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

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

INCOSE. 2010. *Systems Engineering Competencies Framework 2010-0205*. San Diego, CA, USA: International Council on Systems Engineering (INCOSE). INCOSE-TP-2010-003.

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*. Geneva, Switzerland: International Organization for Standardization (ISO). 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*. Geneva, Switzerland: International Organization for Standardization (ISO)/International Electrotechnical Commission (IEC). ISO/IEC: A.4.

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

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

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

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

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

ISO/IEEE. 2008. *Systems and Software Engineering - Software Life Cycle Processes*. Geneva, Switzerland: International Organization for Standards (ISO)/Institute of Electrical & Electronics Engineers (IEEE) Computer Society, ISO/IEEE 12207:2008(E).

Jain, R., 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, R. and D. Verma. 2007. *A report on curriculum content for a graduate program in systems engineering: A proposed framework*. San Diego, CA, USA: International Council on Systems Engineering (INCOSE). INCOSE-PP-2007-001-01.

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

Kelly, M.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.

Kline, S.J. 1995. *Conceptual Foundations for Multidisciplinary Thinking*. Stanford, CA, USA: Stanford University Press.

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, A.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 A. 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.

Likert, R. 1932. "A technique for the measurement of attitudes." *Archives of Psychology*. 22(140): 55.

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

Matchett, G.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, UK: Her Majesty's Stationary Office (HMSO), Cmnd. 6018.

Mooz, H., K. Forsberg, and H. Cotterman. 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 Education* 5 (2) (June): 105-8.

NASA APPEL. 2009. *NASA’s Systems Engineering Competencies. US National Aeronautics and Space Administration (NASA)* Academy of Program/Project & Engineering Leadership (NASA APPEL website. Accessed on September 15, 2011.

http://www.nasa.gov/pdf/303747main_Systems_Engineering_Competencies.pdf

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.

Reitman, W.R. 1962. Systems engineering in interdisciplinary curricula - the program at Carnegie Institute of Technology. *IRE Transactions on Education* 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 Education* 5 (2) (June): 113-5.

Sosniak, L.A. 1994. “The taxonomy, curriculum, and their relations.” In *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, p. 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*

Engineering. 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 Engineering* 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*. 2(3): 184 – 201.

Squires, A., J. Wade, P. Dominick, and D. Gelosh. 2011. "Building a Competency Taxonomy to Guide Experience Acceleration of Lead Program Systems Engineers." Paper presented at the Conference on Systems Engineering Research (CSER) 2011, 15-16 April 2011, Los Angeles, CA.

Truxall, J.G. 1962. “Feedback and systems engineering.” *IRE Transactions on Education* 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).

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

US Office of Personnel Management (OPM), Human Capital Assessment and Accountability Framework (HCAAF) Resource Center, "Glossary". [1] Last accessed June 19, 2012.

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.

Wasson, C. 2012. “System Engineering Competency: The Missing Course in Engineering Education.” Paper presented at the American Society for Engineering Education (ASEE), San Antonio, Texas, June 10 – 13.

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

University of Wisconsin-Madison. 2009. “Outcomes Assessment: Office of the Provost” University of Wisconsin-Madison website. Last modified November 25 2011. Accessed December 4, 2012. Available at: 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. 2010. *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.

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

A.1 Background

The GRCSE author team is comprised of 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 extended 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 is derived from the principles stated in 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 in that 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 also 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 Systems Engineering

According to the Guide to the Systems Engineering Body of Knowledge (SEBoK), SE is:

> *[A]n interdisciplinary approach and means to enable the realization of successful systems. It focuses on holistically and concurrently understanding stakeholder needs; exploring opportunities; documenting requirements; and synthesizing, verifying, validating, and evolving solutions while considering the complete problem, from system concept exploration through system disposal (Pyster and Olwell (eds.) et al. 2012).*

SE as a discipline can be applied in a variety of settings and the principles of SE can be used to improve work flow in every-day tasks as well as in major engineering projects.

A.3 Assumptions

There are 19 statements that guided the development of GRCSE. This section presents four assumptions about SE that have shaped the development.

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 address SE. The SEBoK is the SE community's guide to its rich body of knowledge. Direct linkages to the SEBoK are established throughout GRCSE, specifically in Chapter 3 (outcomes) and Chapter 6 (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, project management, economics, systems science, social sciences, and 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 also there are wide variations 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 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.4 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; however, their ordering does not imply importance or precedence.

A.4.1 Scope

5. *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 may be used as an input to organizations which develop and maintain standards for accreditation, certification, and licensure.

6. *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.

7. *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 promote community consistency in graduate SE education.

Curricular requirements and educational expectations differ between countries, but the issues 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.4.2 Program Architecture

8. *A master's program that satisfies the GRCSE recommendations should require approximately the same amount of study as programs do now.*[9]

   The intent is for GRCSE to fit within the common scale of current 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).

9. *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. The knowledge and experience an entrant obtains from undergraduate programs and industry vary greatly. However, to help institutions build programs that address the needs of the broad SE community, GRCSE

[9] 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 to take proportions which meaningfully indicate amounts of effort devoted to particular parts of a program by reference to a proportion of a whole program.

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.

10. *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 conceptual program architecture used to frame all discussions in GRCSE.

11. *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 a typical master's degree, reserving time for university specific content. This provides program flexibility for defining ways to achieve its outcomes and objectives.

    GRCSE expects that curriculum developers will design their programs in a manner that is cognizant of the actual preparation of their typical entrants, as well as that they will develop teaching and learning experiences that enable the students admitted to the program to develop the intended knowledge and skills through the program.

A.4.3 Context

12. *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.

13. *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. To assist, Chapter 7 of GRCSE provides guidelines for its implementation.

14. *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 a recognition of the need for other expertise in a rapidly changing discipline. These are professional attributes that ensures the systems engineer can deliver suitable solutions to satisfy all required needs. These issues and activities are addressed in the objectives (Chapter 2), outcomes (Chapter 3), and the CorBoK (Chapter 6).

15. *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; however, 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 component of this involves 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 discussion of outcomes (Chapter 3).

16. *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 highly technical skills to "soft skills", which involve communication, ethics and other 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 that is critical for graduates of SE master's programs is defined within the CorBoK (Chapter 6).

17. *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 of these employer bases.

18. *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.*[10]

    Appendix C 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.

19. *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 for further information.)

[10] 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.

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 SE master's degree (MSE) programs. Survey forms were sent to 59 schools, which offered a combined total of 63 systems-centric systems engineering programs. The initial list of candidate U.S. 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 United States. Survey responses were received for 33 programs from the following countries: Australia, Brazil, Canada, Cyprus, Finland, France, Japan, Singapore, Taiwan, and the United States. The range of schools that responded to the survey included those from traditional universities and specific-purpose institutions (such as the postgraduate institutions and alliances). It is important to note that the survey data was collected on condition of anonymity; the schools were asked to provide honest responses and in return, the author team agreed not to publish the names of the specific institutions responding. It is also important to note that, while SE is often incorporated in some ways into other engineering programs (e.g. mechanical, electrical, material, civil), this survey specifically focused on SE master's programs.

A survey instrument, similar to the one shown in Table 17 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 that the data was analyzed. The analysis can be found in Section B.2, below.

Table 17. Survey Data Collection Instrument. (GRCSE Original)

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 hour.)

B.2 Findings

It is important to note that the survey responses further illustrated a high level of variation found in these programs. The spectrum of programs investigated led to a number of findings about student acceptance criteria, completion criteria, general focus, and core courses of the programs. The initial findings of the survey are presented in Table 18. 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 18. Survey Results. (GRCSE Original)

Program Attributes	
Delivery Mode	**% Schools**
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. Those categories are represented here. Note that the percentage for "Top 10 Courses" is the percentage of respondents with a course of this title; therefore, the percentages do not add up to 100%.

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

- The findings show that 71% of non-U.S. programs and 33% of U.S.-based programs used only face-to-face interaction in the delivery of their courses. In addition, 67% percent of U.S. based institutions utilized 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. Please note that the results in Table 18 report all programs using each technique; the categories are not mutually exclusive.
- 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 universities preferred that incoming students have an engineering degree (88%), though many would allow students with a technical, non-engineering degree into the program (81%). Only 30% of programs surveyed would allow students into their programs with non-technical, non-engineering degrees.
- Most U.S. based schools surveyed required an undergraduate grade point average (GPA) of 3.0 or better. In contrast, 79% of the non-U.S. 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 U.S., were not generally required outside of the U.S. Only one university that was not based in the U.S. 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 U.S. based schools that accept international students. In addition, the TOEFL and IELTS were requirements by non-U.S. based schools; they specifically identified 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.

Appendix C: Bloom's Taxonomy of Educational Outcomes

This appendix provides a general discussion of Bloom's taxonomy of educational outcomes as a foundation for the use of Bloom's taxonomy in other parts of GRCSE. The major place where Bloom's taxonomy and levels of attainment, commonly referred to as "Bloom levels", are used in GRCSE is in the description of the 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. The original exposition of the taxonomy, contained in Bloom et al. (1956) and Krathwohl, Bloom, and Masia (1964) remains the best source for explaining the 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 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 1950's. 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 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). The hierarchical and behaviorist perspective is suited to an education program, such as a professional master's program, the focus of GRCSE, where graduates are expected to be able to perform a certain range of tasks with a defined level of competency. This differs from the matters of interest in other fields of education.

Systems engineering is a field of professional practice which is effectively described by the relationships of a framework of ideas, with F in Figure 1, describing the knowledge required for practice. This framework of ideas is applied to particular concerns in the performance of SE, which is the area of concern and application, represented as A in Figure 7, using the methodology M_1. The methods, M_1, to apply the framework of ideas of systems engineering are contained within the framework of ideas. Good professional practice should involve reflection upon the effect of the work done in a particular application of the framework of ideas, in a particular application of SE, which in turn could improve the framework of ideas. This reflective practice is shown by M_2 in Figure 7. In addition, research methods which are suitable for improving the framework of ideas of SE are available to enable research projects, M_3. Figure 7 also shows the role of education in SE to develop students' knowledge of the framework of ideas and the methods to perform systems engineering. Education to support both these aspects of professional systems engineering practice is described in GRCSE. Education is also required to enable learning of how to perform research in systems engineering, which is out of scope of GRCSE.

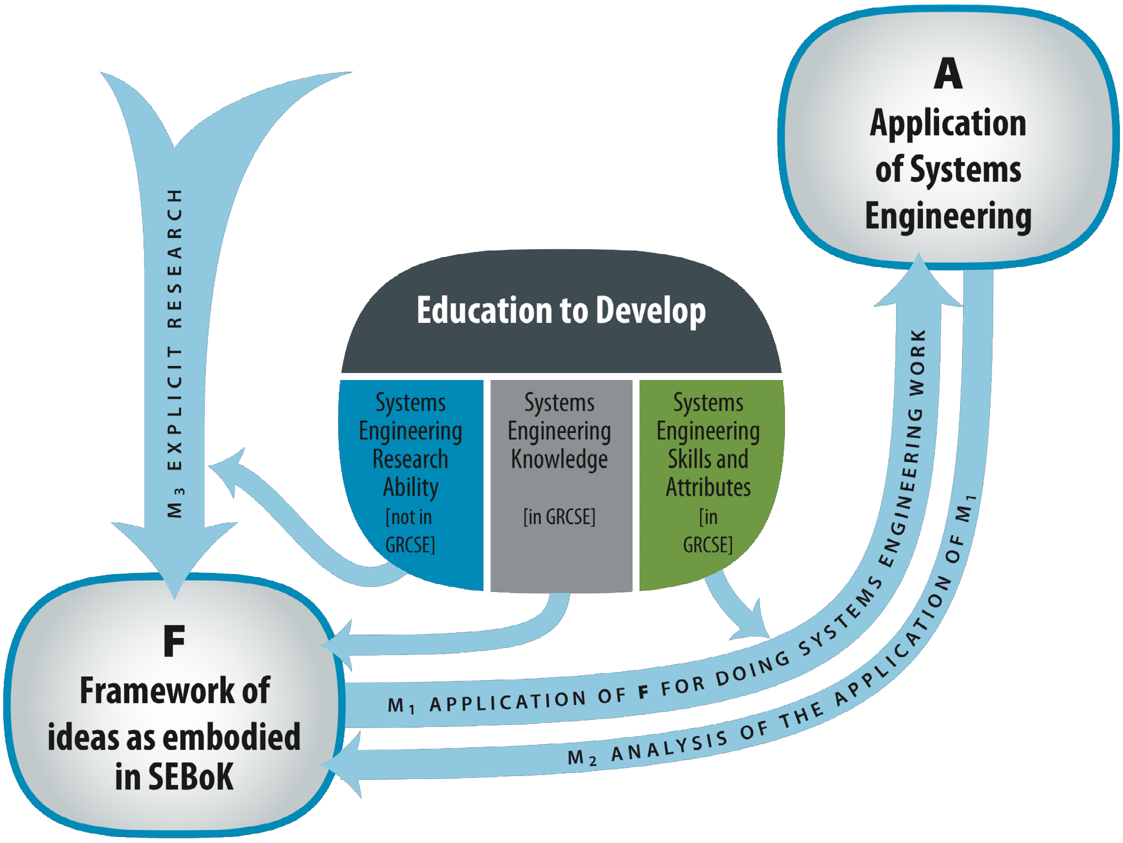


Figure 7. The characteristics of systems engineering as a professional field of practice and the place of GRCSE in supporting education which enables the capabilities required. (GRCSE Original)

Bloom's taxonomy divides educational outcomes into three domains: cognitive, affective, and psychomotor. The original 1956 and 1964 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 of information and methods, the framework of ideas of SE, 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). These two domains, their hierarchical structure and relative emphasis are shown in Figure 8. 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. The CorBoK, as presented in Chapter 6, emphasizes the cognitive domain. In addition to showing the hierarchy of levels of achievement in the cognitive and affective domains, Figure 8also shows the relationship of the levels of achievement in the cognitive domain and the levels of competency as described in the INCOSE competency framework (INCOSE 2010). In turn this illustrates, comparing the level of achievement in most areas recommended in the CorBoK and the levels of competency, that it is reasonable to expect a new graduate in SE to be able to perform work as a supervised practitioner. However, that graduate, by the time that they achieve the objectives of their program would be expected to grow in competency to be an independent practitioner.

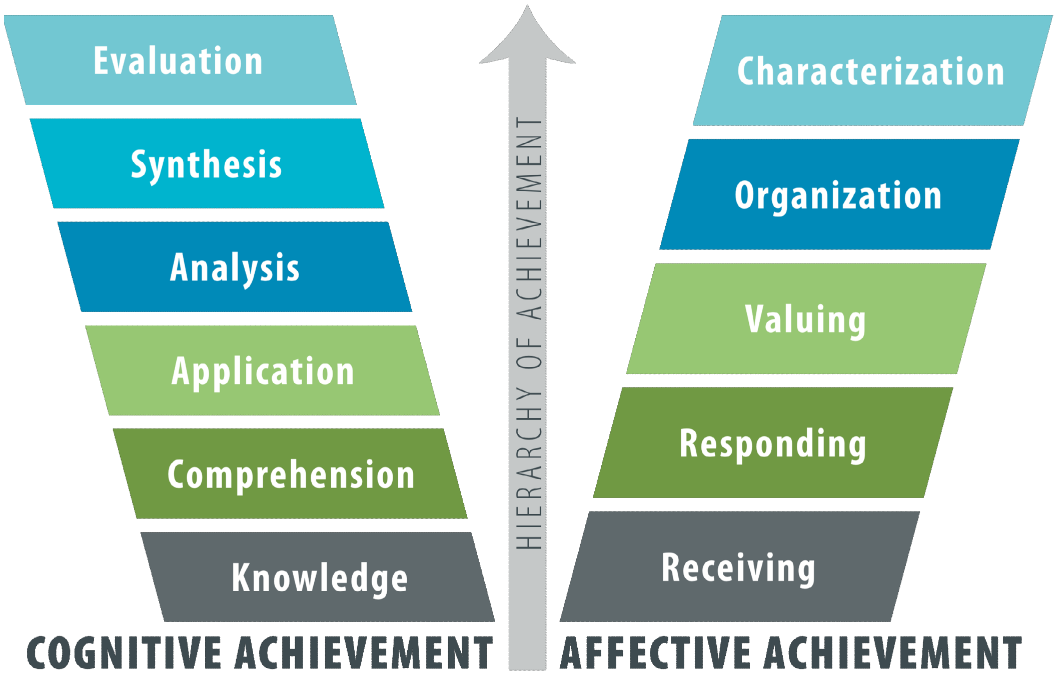


Figure 8. Visual representation of the hierarchical nature of Bloom's taxonomy in the cognitive and affective domains and the relationship of the levels of achievement with competency levels. (GRCSE Original)

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.4.3, Context Guiding Principle #7), and explicitly discusses the relationships of the affective domain to SE education in this appendix. A key 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 of a systems engineer make their proficiency in the attributes of the affective domain critical to their success. This is not only obvious in relation to the competencies associated with teamwork and leadership of projects, but also has a more subtle manifestation. 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 matters in the whole system context. The negotiation role involves making judgments about what is good or desirable and also demands that systems engineers personally value the systems perspectives required to perform their work.

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

Table 19 provides a description of Bloom's taxonomy for the cognitive domain. The levels of the taxonomy are commonly referred to as Bloom's levels. In a general context, the term "Bloom's levels" 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 as to 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; ergo, in regard 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 reflect 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 of performing successfully in a workplace setting as well as to reflect the 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 good judgments about what design is appropriate for a particular application. That is, the action of synthesis is prior to evaluation in engineering work and defines the level of acuity required to perform work. To demonstrate how these levels may be applied specifically to SE, Table 20 shows examples of various Bloom cognitive domain level competencies that might apply to GRCSE curricula and courses in the form of possible assignments students could complete.

In discussing the hierarchical nature of Bloom's taxonomy it is important to be thorough and clear in describing 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 simply addressing 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 attained by the same type of studying that achieved the original knowledge; rather, it is achieved by studying and performing tasks that redirect the student from activities that are required to develop knowledge to those that are required to develop comprehension. Similarly, "Analysis" and "Synthesis" are different kinds of skills that involve different approaches 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 based on a different kind of learning.

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 19. Explanation of Bloom Taxonomy Cognitive Levels. (GRCSE Original)

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.
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 the identification of parts.	Analyze, separate, order, explain, connect, classify, arrange, divide, compare, select, explain, and infer.

Level	Sub-Level	Competency	Outcome Descriptors
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 building upon 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 20. Example of Cognitive Levels for Systems Engineering. (GRCSE Original)

Level	Example Competencies and Assessment
Knowledge (K)	• The student can recite the definitions of "system" and "emergence" and state the connection between them. • The student can describe the notion of product system architecture, in a manner consistent with commercial standards, and state in general terms the impact that architecture may have on system development success.
Comprehension (CO)	• The student can explain in his own words the conditions under which a system development team might choose to use a specific life cycle model, such as iterative, incremental, or spiral. • The student can explain in his own words the range of cases for which a particular systems modeling approach is applicable.
Application (AP)	• The student can select the appropriate life cycle model for a project given project parameters such as budget, deliverables, schedule, and requirements. • The student can develop a technical management plan using specific planning methods when given the project parameters such as budget, deliverables, and requirements. • The student can use a specific technique to model a system defined and described in a task description.

Level	Example Competencies and Assessment
Analysis (AN)	• The student can provide a detailed critique of the choice of a life cycle model made by someone else for a project with specified budget, deliverables, schedule, and requirements. • The student can fully justify the approach and details of a technical management plan the student previously developed, including rationale for the selection of planning methods and tools. • The student can determine how well application of a particular modeling technique will produce the desired insights into a system's behavior, including circumstances under which the model might lead to misleading or erroneous insights
Synthesis (S)	• The student can take life cycle models used to create individual subsystems and develop a new hybrid life cycle model for the larger system that integrates and modifies elements of the lesser life cycle models as appropriate. • The student can draw from a set of smaller systems with which he is familiar and develop an architecture for a larger system that satisfies a set of specified requirements and uses those smaller systems as the primary building blocks for the larger system. • The student can develop and use a variety of modeling techniques in concert to analyze and predict the behavior of a system and how well those techniques will work; e.g., using a combination of physics-based behavioral models to determine the fuel economy of an automobile. • The student can develop a verification and validation plan for a system which has subsystems that require significantly different verification and validation approaches (e.g., a system which has pure software, legacy hardware, and newly developed software/hardware subsystems) and argue why the plan will be effective and what its limitations will be.

Level	Example Competencies and Assessment
Evaluation (EV)	• The student can explain why they chose a specific combination of modeling techniques to analyze and predict behavior, what the limitations of that combination are, what the limitations of other combinations would be, and suggest areas in which those modeling techniques should be improved. • The student can explain the relative merits of different architectural approaches in the context of real architectural examples, what the limitations of those architectural approaches are, and suggest areas in which those architectural approaches could be improved. • 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 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.

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

Table 21 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). In particular, systems engineers need to appreciate the subtleties of interaction with colleagues in order to elicit the best effect from teamwork and to be characterized by their belief that taking a holistic and systemic approach to engineering is the best approach for achieving successful results. To demonstrate how these levels may be applied specifically to SE, Table 22 shows some examples of various Bloom affective domain competencies that might apply to GRCSE curricula and courses.

Table 21. Explanation of Bloom Taxonomy Affective Levels. (GRCSE Original)

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, demonstrates alertness 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 a strong interest in aesthetics.	Demonstrates willing compliance and obedience to regulations and rules, seeks broad-based information to act upon, and accepts responsibility and expresses pleasure for one's own situation.
Valuing (V)	• Acceptance of a value • Preference for a value • Commitment	The learner recognizes worth in the subject matter.	Shows continuing desire to achieve, assumes responsibility for, seeks to form a view on controversial matters, displays devotion to principles, and demonstrates 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 22. Example Affective Levels for SE. (GRCSE Original)

Level	Example Competencies	Possible Assessment Tasks

Level	Example Competencies	Possible Assessment Tasks
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.	• An assignment to explain how customer or user perception of the system governs recognition of quality of the system. • An assignment to explain the challenges in eliciting needs and/or requirements in a case study project. • An assignment to the student to describe the financial value of SE works in projects.
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.	• An assignment to interview stakeholders of a project concerning the needs/requirements for the system under development. • A project for which SE methods are demanded with the use of a reflective journal which requires discussion of the usefulness of SE methods in doing the 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 the proficiency of SE competencies.	• An assignment to show the value to a system developer's future business from the reputational effect of properly attending to stakeholder needs. • An assignment in which the student analyzes the impact on a case study project where there is evidence that needs/requirements elicitation were significantly inadequate in capturing the real interests of the stakeholders. • A task to provide a defense of use of SE methods, based on the information available at the time, in the early stages of a case study project which used some other method. • A task analyzing the SE competencies and levels of attainment that are required for particular SE roles within a large project.
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	• An assignment to propose and justify, through explaining the expected benefits, the use of particular SE methods and processes for a particular project. • A project with "external" (to the academic department) stakeholders

Level	Example Competencies	Possible Assessment Tasks
	the views of a variety of stakeholders in a balanced manner.	where the student must justify the method used to perform the project in terms of the expected benefit to the stakeholders.
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.	• A practical examination requiring the development of a system concept where there is a tempting solution, that is obvious given the student's background, designed to test what the student does under time pressure. • A project task in which there is significant conflict between the stakeholders, and assessing a combination of the result delivered by the student and a reflective journal about the process of resolving the conflicts.

Appendix D: GRCSE Outcomes to CorBoK Mapping

Chapter 3 introduces a set of recommended outcomes for the graduates of a SE program that follows the GRCSE recommendations. These outcomes represent the abilities of students at the time of their graduation. Chapters 5 and 6 present a recommended curriculum architecture and CorBoK, respectively, that represent about 50% of the curriculum for a SE program using GRCSE. The content of the curriculum plays a major role in student's achievement of the outcomes. The following table represents the relationship between the CorBoK knowledge areas and each outcome. A knowledge area can support the achievement of an outcome in one of three ways:

M: The specific knowledge area will **moderately** support this outcome.

S: The specific knowledge area will **strongly** support this outcome.

: A blank cell indicates the specific knowledge area will **weakly** support this outcome.

For example, in the case of SE Concept outcomes (FOUNDATION, CONCENTRATION, and TOPIC DEPTH outcomes), each knowledge area in the CorBoK will at least moderately support each outcome. However, depending on the nature of the program (i.e., the Core Concentration selected and/or a university's customization based on its constituency), some of these knowledge areas will strongly support one or more of the SE Concepts outcomes.

Table 23. Alignment of GRCSE Outcomes and the CorBoK. (GRCSE Original)

	GRCSE Outcome												
	SE Concepts			SE Role				SE Practice			SE Professionalism		
CorBoK Part/Knowledge Area	Foundation*	Concentration*	Topic Depth*	Application Domain	Specialty	Related Disciplines	Software in Systems	Requirement reconciliation	Problem/Solution Evaluation	Realism	Professional Development	Teamwork	Ethics
Part 2													
Systems Fundamentals	M/S	M/S	M/S	M			M	M	M	M			
Systems Science	M/S	M/S	M/S	M				M	M	M			
Systems Thinking	M/S	M/S	M/S	M	M	M		M	M	M			
Representing Systems with Models	M/S	M/S	M/S	S	S	M	M	S	S	M			
Systems Approach Applied to Engineered Systems	M/S	M/S	M/S	M		M		S	S	M			
Part 3													
Life Cycle Models	M/S	M/S	M/S	S	S		M	S	S	S			
Concept Definition	M/S	M/S	M/S	M	M	M		S	S	S			
System Definition	M/S	M/S	M/S	M		M	M	M	M	S			M
System Realization	M/S	M/S	M/S	S	S		M	S	S	S		M	
System Deployment and	M/S	M/S	M/S	M	S	M	M	M	S	S		M	M

CorBoK Part/Knowledge Area	GRCSE Outcome												
	SE Concepts			SE Role				SE Practice			SE Professionalism		
	Foundation*	Concentration*	Topic Depth*	Application Domain	Specialty	Related Disciplines	Software in Systems	Requirement reconciliation	Problem/Solution Evaluation	Realism	Professional Development	Teamwork	Ethics
Use													
SE Management	M/S	M/S	M/S	M	M		M	M	M	M		S	M
Product and Service Life Management	M/S	M/S	M/S	M				M	S	M		M	M
SE Standards	M/S	M/S	M/S	S	S			M	S	M			S
Part 4													
Product SE	M/S	M/S	M/S	M	M	M	M	M	M	M			
Service SE	M/S	M/S	M/S	M	M	M	M	M	M	M			
Enterprise SE	M/S	M/S	M/S	M	M	M	M	M	M	M			
Systems of Systems (SoS)	M/S	M/S	M/S	M	M	M	M	M	M	M			
Part 5													
Enabling Businesses and Enterprises	M/S	M/S	M/S					M			S	S	M
Enabling Teams	M/S	M/S	M/S	S				M	S	M	S	S	S
Enabling Individuals	M/S	M/S	M/S	S				M	S	M	S	S	S
Part 6													
SE and Software Engineering	M/S	M/S	M/S	M	S	S	S	M	M	M			
SE and Project Management	M/S	M/S	M/S	M		S		M	M	M		S	
SE and Industrial Engineering	M/S	M/S	M/S	M	S	S		M	M	M			
SE and Procurement/Acquisition	M/S	M/S	M/S	M	S	S				M			M
SE and Specialty Engineering	M/S	M/S	M/S	M	S	S		M	M	M			

Appendix E: Assessment and Achievement of Learning Outcomes

This appendix presents additional, complementary material for Chapter 8. It describes a general framework for assessment that is independent of the domain covered by the program as well as its academic level (undergraduate or postgraduate). This framework is a generic reference that should be used to drive the definition and execution of the 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 how to apply that framework to assess systems engineering master's programs.

E.1 Purpose, Scope and Justification of Assessment

The purpose of assessment is to evaluate whether learning outcomes and objectives are being achieved to improve the learning process (formative assessment) or to judge the result of the learning process (summative assessment).[11]

Assessment must be seen as a way to continuously obtain feedback from various stakeholders involved in the learning process (e.g. students, teachers, employers) 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).

The scope of assessment can encompass a variety of targets, 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 outcomes and assessment of program objectives.
- **Institutional Assessment**: Involves assessment of campus-wide characteristics and issues.

The logical relationships of the various targets of assessment are shown in Figure 9. It is important to notice that these levels are not independent, as 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.

[11] Our definition encompasses ABET definitions of assessment (one or more processes that identify, collect, and prepare data to evaluate the achievement of program outcomes and program educational objectives) and evaluation (one or more processes for interpreting the data and evidence accumulated through assessment processes) (ABET 2010, 1).

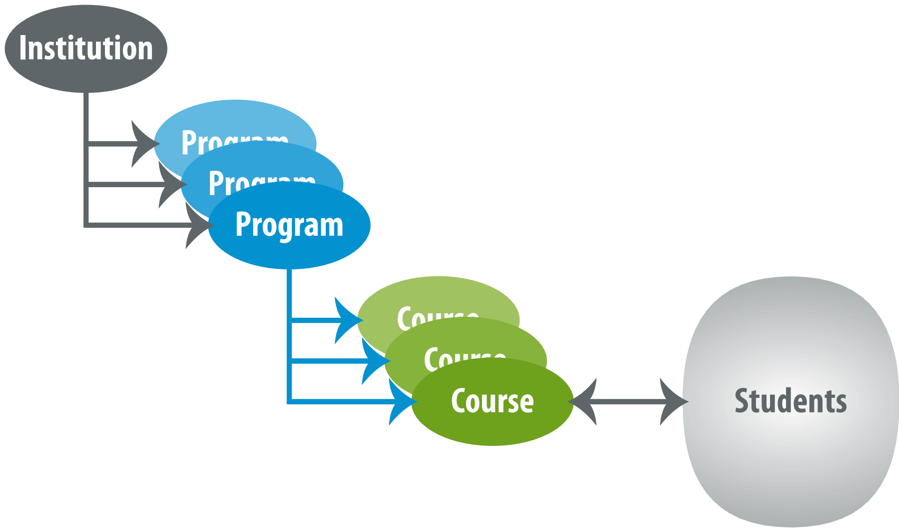


Figure 9. Various Targets for Assessment. (GRCSE Original)

The justification of assessment in a program has been explained by Gray (1977), who described the educational paradigm in what he called the "training triangle," which consisted of three points: outcomes, teaching methods, and assessment, as illustrated in Figure 10. He demonstrated that assessment is "an integral part of the educational process" as reported by the Merrison Committee (1975). Without assessment, students will not know how or whether they benefited from their education, 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.

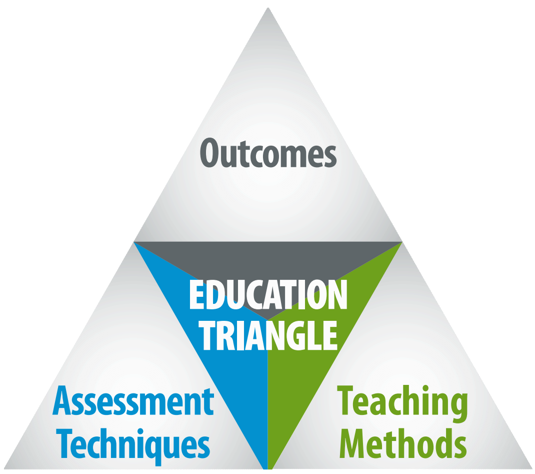


Figure 10. Training Triangle (Based on Principles from Gray 1977). (GRCSE Original)

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.2 Assessment Process

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

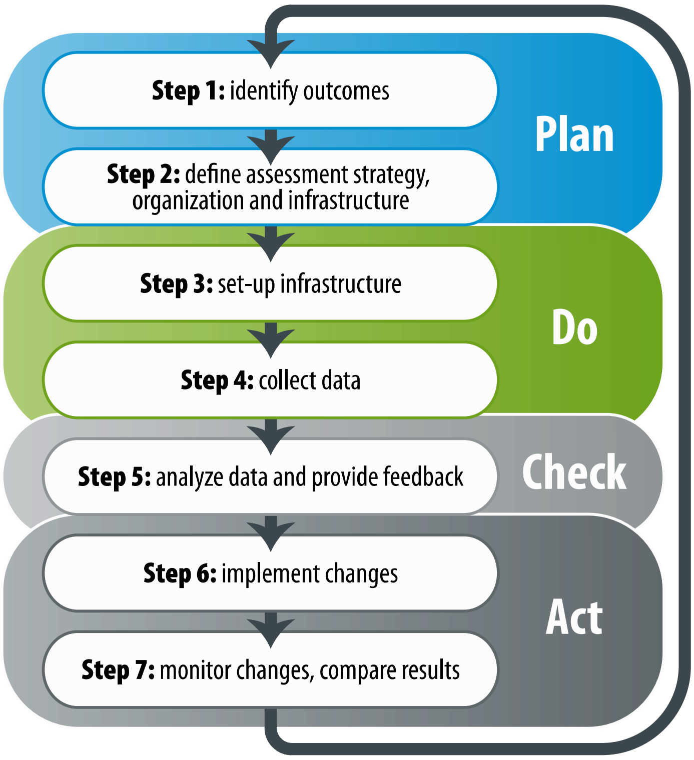


Figure 11. Assessment General Process. (GRCSE Original)

Step 1: Identifying Outcomes and Student Expectations

The objective of step 1 is to identify required outcomes, or what the students are expected to know and perform throughout the course or program using the CorBok. GRCSE, as well as many instructors, use Bloom's taxonomy (Bloom et al. 1956) to describe levels for learning achievements. There are more recent works by Gronlund (2000) and Krathwohl (2002) also used as alternatives. Training evaluation models, such as the Kirkpatrick model (Kirkpatrick 1994), are primarily used today by industry and considered 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 (or learning modules) that constitute the course. The lesson outcomes should clearly point back to and support the achievement of the course objectives; the course objectives should clearly point to program objectives; and the program objectives to the overall college or university's general education objectives.

Outcomes should be measurable. 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, as it provides a capacity for determining if a student fulfills all program outcomes and is prepared to begin his or her career. Performance criteria are compared to the targets assessed that must be achieved for graduation.

Step 2: Define Assessment Strategy and Methods

Step 2 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:

- The outcomes for assessment.
- How the assessment will be completed (i.e., by selecting appropriate assessment methods).
- The organization of the assessment - identifying task, planning, and resources.
- The infrastructure required to perform task assessment.

Step 2.1: Define Assessment Methods

The objective of step 2.1 is to select the outcome and allocate it to an assessment method. This selection is an essential step to ensure the success of the assessment process. A good assessment method should consist of the following characteristics:

- **Validity**: Validity refers to determining whether the selected assessment method is appropriate for measuring what it is intended to measure (e.g. content, Bloom's level).
- **Reliability**: Reliability yields consistent responses over time. To ensure that the assessed objective be reproducible –the same inputs should provide same results.

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. 2011), it is necessary to gather evidence to support the interpretation and appropriateness of a survey or test for a specific purpose. For example, pertinence of performance criteria, data source representation, and the possibility of a student cheating should be carefully studied and the appropriate actions taken (e.g. the use of multiple data sources and assessment methods).

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 may also 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 prioritizing outcomes. The assessment method(s) should be as precise as needed by the priority of the assessed outcome.

This information may be displayed in a table indicating which outcomes each assessment method will measure. It is not necessary to assess all of the outcomes at once. An initial focus on those most important is preferred (i.e., the outcomes for the data that will be most useful to the program). In some instances, one assessment method may measure multiple outcomes, and one outcome may be measured by multiple assessment methods.

Table 24 is an inventory of the major or most frequently used assessment methods, shown as "tools" in the table (Fulks 2006), including supporting information for selecting the methods. These methods rely on feedback from diverse populations (e.g., internships, supervisors, student self-reports, employers, and teachers) and are categorized as either direct or indirect in the table. 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 testing performance, review of student work (e.g., portfolios or theses), or observation of student performance within appropriate context. Indirect methods differ because they deal with students' experiences, opinions, or perceptions. They can be self-reported by the student or obtained by faculty or employers. In general, direct measures are strongly preferred during assessment, but may be impractical when assessing program objectives. For a systematic presentation and analysis of assessment methods, refer to (Prus 1994) or (UCF 2008).

Table 24. Potential Methods of Assessment (Fulks 2004). Released by Janet Fulks, Assessing Student Learning in Community Colleges (2004), Bakersfield College.

Assessment Tool	Direct or Indirect (D or I)	Bloom's Level (s)	Pros	Cons
Multiple Choice Exam	D	Knowledge, Comprehension	• Easy to grade • Objective	• Reduces assessment to multiple choice answers
Checklists	D	Variable	• 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	All	• Good display of analytical and synthetic thinking	• Time consuming to grade, can be subjective
Case Study	D	All	• 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	All	• 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	Variable All	• 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
Product Creation & Special Reports	D	Variable All	• 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

Assessment Tool	Direct or Indirect (D or I)	Bloom's Level (s)	Pros	Cons
Portfolios	D	Variable	• 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 evaluation difficult and may require training • Bulky to manage depending on size
Exit Surveys	D, I	Analysis, Synthesis, & Evaluation	• 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
Capstone Project or Course	D	Analysis, Synthesis, & Evaluation	• 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
Satisfaction and Perception Surveys	I	Comprehension, Application, Analysis, Synthesis, & Evaluation	• 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. Special attention should be paid to the selection of the sample group (students, employers, etc.)

Step 2.2: Define the Life Cycle of Assessment

Step 2.2 of the strategy defines when an assessment is performed during a course or program. When performed at the beginning or in the middle of the course, the assessment provides feedback that is directly usable by students and teachers concurrently with course delivery, so results can be formative and influential. When performed at the end of the course, assessment measures the final learning achievement for grading purposes and serves as a way to improve the course. Therefore, a final assessment can be both summative and formative. Many programs conduct both mid-course and end-of-course assessments.

Step 2.3: Define Infrastructure to Support the Assessment Process

Step 2.3 defines the tools and input data that are required to perform the assessment. 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, like a set of test questions. When a survey of employers is selected, it is important to maintain at least a set of the employers over time.

Step 2.4: Define the Organization for Assessment

The objective of Step 2.4 is to perform project management activities related to the assessment process by defining tasks, identifying inputs and outputs for each task, identifying planning, and forecasting the workload necessary to perform the assessment tasks. Organization should also focus on the people involved in the assessment process by defining the roles, identifying the people to support roles, and defining the recruitment and training needs.

It is important to perform assessment through diverse stakeholders involved in the program (e.g, students, teachers, employers, and former students). This can also take the form of a strategic advisory board. External assessors should be educated in systems engineering and should have a clear understanding of the university's approach. Diversity of stakeholders enables the reception of many viewpoints as well as a high capacity for curriculum evolution and development.

Step 3: Set Up the Infrastructure

Step 3 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

Step 4 is the execution of the assessment specified in Step 2.3 and then realized in Step 3.

Step 5: Analyze Collected Data and Provide Feedback

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

- **Student sub-populations:** In the case of summative assessment the data is 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. It also identifies areas of strength and weakness for students, as well as ways to improve 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 functionality 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, such as, changing which data is collected as well as data collection methods (see Step 6). 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:

- **Assessment Strategy**: Changes to assessment strategy may include a revision of assessment methods and their relationship to outcomes, the collection and analysis of additional data and information, or changes to the data collection methods.
- **Program Outcomes**: Changes to program outcomes may include a revision of intended learning outcome statement(s) and their priorities.
- **Curriculum**: Changes to curriculum may include changes in learning methods, a revision or enforcement of expected student backgrounds, a revision of course sequences, a revision of course content, and/or the addition or deletion of course(s).
- **Academic Processes**: Changes to academic processes may include the modification of the frequency or scheduling of course offerings, improvements in technology, changes in personnel, implementation of additional training, other implemented or planned changes, a 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; for example, 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 systems engineering graduate program faculty.

Step 7: Monitor Changes and Compare Results

Step 7 monitors the implemented changes as part of the continuous improvement cycle. The implemented changes are monitored to determine whether they have achieved the desired effect(s).

One way to achieve this is to use the same assessment plan as used in the previous cycle and compare the actual data to the intended data. This approach's drawback is that the people doing the assessment will not necessarily be the same. 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 may also be worthwhile to reach out to the people who performed the former assessment, describe the changes implemented, and obtain feedback on the direction of the changes, even though these individuals may not have experienced the changes themselves.

Additionally, the program environment is evolving, thus driving the program to adapt so that it continues to provide its students and community with a suitable education.

Any gaps in an assessment strategy should be studied carefully to determine the underlying cause. Where outcomes are met the best course of action is typically to maintain monitoring of them to preserve program quality.

Appendix F: Competency-Based Curriculum Development

F.1 Introduction

GRCSE provides the guidance necessary to create a SE curriculum that supports attainment of the objectives described in Chapter 2 and the outcomes described in Chapter 3. Appendix F offers 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.

This appendix discusses 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 captures the elements of the approach advocated in this appendix; these life-cycle elements are discussed further in the subsequent sections.

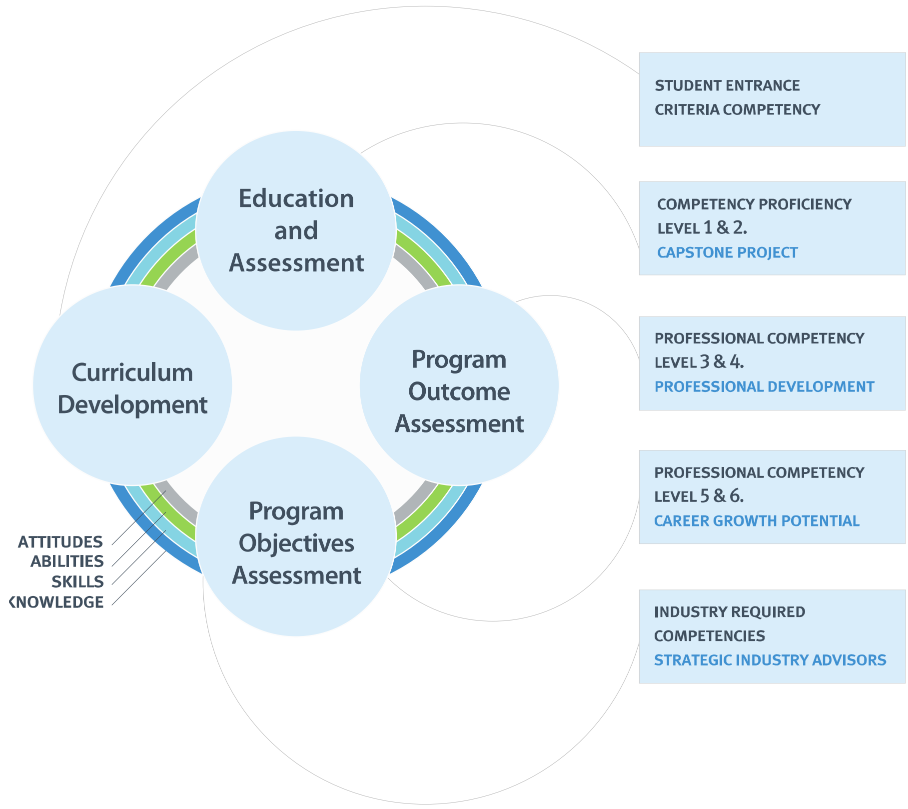


Figure 12. Competency Development Life-Cycle. (GRCSE Original)

F.2 The Basics of Competency Models

The 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 (2010) summarizes and evaluates these models in terms of contribution to the selection of personnel and for informing systems engineering education. Ferris (2010) notes that each model covers technical competencies that extend across domains, in support of the multi-disciplinary nature of SE, as well as that each model addresses behavioral competencies with leadership as a common theme. Wasson (2012) builds the case for the inclusion of SE competency in engineering education, with graduate level education addressing methods for project specific tailoring of SE practices to achieve technical, cost, and schedule objectives.

The term “competency” is used in a number of different contexts. For example, GRCSE defines outcomes, which are effectively expected student competencies upon graduation, and objectives, which are competencies students are expected to achieve three to five years after graduation. 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. In this appendix, the term “competency” is meant to apply to a capability throughout the career of a systems engineer. 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). Notice Figure 12(above) uses the KSAA elements to form guiding rings around the entire competency development life cycle with the curriculum elements distributed around the KSAA rings.

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 (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 – Systems engineering competencies such as project management, requirements management, risk management, integration, and systems architecture.
- Specialization Competencies – Competencies specific to a job type or an application domain (e.g., competencies specific to aviation systems).

Squires et al. (2011) demonstrates the integration of competency models with a competency taxonomy that guides experience acceleration of lead program systems engineers using simulation and gaming technology. The taxonomy addresses commonalities between existing SE competency models in six broad categories of competencies:

1. Systems Thinking and Critical Thinking as the backbone or hub of SE competency.
2. Technical Leadership at the enterprise level.
3. Technical Management across the system life cycle.
4. Technical/Analytical implementation of SE at the practitioner level.
5. Project Management as the domain independent practice that also supports SE management.
6. Other broad and professional competencies related to leadership and development, as well as program assessment and recovery.

Table 25 provides another view of how competencies may be categorized. Of course, these or other approaches could be used to organize and describe an organizations' competency model.

Table 25. Competency Categories. (GRCSE Original)

	Technical Competencies	Cognitive Competencies	Behavioral Competencies
Definition	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.
Examples	• 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:

- Ability to identify all the stakeholders and their sphere of influence.
- Ability to elicit requirements from stakeholders.
- Ability to create a traceability matrix of requirements for design and implementation.
- Ability to create a set of requirements that meets the quality specifications of writing a good requirement.
- Ability to define a process to manage the requirements and ensure their effective implementation.

- Ability 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 "behavioral" 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. Table 26 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. 2011)[12].

Table 26. Competency Proficiency Levels (VanLeer 2007). Permission Granted by Mary VanLeer. All other rights are reserved by the copyright owner.

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 27. Competency Proficiency Levels (Squires et al. 2010). (GRCSE Original)

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.

Note that in Table 26 there are five different levels of competency proficiency distributed throughout the competency development life cycle. This is also demonstrated in Figure 12 (above):

- At the first level in Figure 12 there is the "Student Entrance Criteria Competency." This proficiency level is discussed in Chapter 4.

[12] From "Building a Competency Taxonomy to Guide Experience Acceleration of Lead Program Systems Engineers," which was presented at the Conference on Systems Engineering Research (CSER).

- Proficiency levels 1 and 2 are associated with the attainment of the student outcomes for a GRCSE-aligned 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 overlap with attainment of the objectives associated with a GRCSE-aligned curriculum, which are achieved in the first years of SE professional practice.
- Proficiency levels 3 and 4 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

Table 28 offers a process that provides guidance on developing a competency-based curriculum. The process combines the recommendations offered by GRCSE together with an understanding and appreciation of professional career development. The resulting curriculum recognizes that competency development must extend throughout an individual's career, as well as what competency models play a key part in assessing and advancing professional competency.

Table 28. Process Script for Competency-based Curriculum Development. (GRCSE original)

Process Step	Description
Step 1: Establish the Development Team	• Identify program stakeholders • Form team from stakeholders such as faculty responsible for curriculum design and representatives from prospective employing organizations. • Use a wide group of reviewers for each stage of development.
Step 2: Create Competency Model	• Study and analyze various competency models. • Choose or adapt competencies appropriate to the stakeholders. • Define proficiency levels. • Select target proficiency levels for each competency.
Step 3: Prepare Draft Curriculum	• Based on competency model, prepare draft outcomes and objectives. • Use outcomes and objectives to prepare a draft curriculum.
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 CorBoK, or individual course activities. • Determine required changes in the curriculum and implement them. • Modify the competency model based on the competency assessment report. • Repeat steps 4 and 5 after implementation to ensure continual evolution and improvement of the curriculum.

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

Steps 3 and 4 require activities beyond GRCSE guidance. VanLeer (2007), Squires and Larson (2009) and 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 (Squires and Larson 2009). (GRCSE Original)

Process Step	Description
Step 1: Establish the Development Team	• Identify program stakeholders • Form team from stakeholders such as faculty responsible for curriculum design and representatives from prospective employing organizations. • Use a wide group of reviewers for each stage of development.
Step 2: Create Competency Model	• Study and analyze various competency models. • Choose or adapt competencies appropriate to the stakeholders. • Define proficiency levels. • Select target proficiency levels for each competency.
Step 3: Prepare Draft Curriculum	• Based on competency model, prepare draft outcomes and objectives. • Use outcomes and objectives to prepare a draft curriculum.
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 CorBoK, or individual course activities. • Determine required changes in the curriculum and implement them. • Modify the competency model based on the competency assessment report. • Repeat steps 4 and 5 after implementation to ensure continual evolution and improvement of the curriculum.

Appendix G: Use Cases

It is the intention of the GRCSE authors that GRCSE be useful to the global SE educational community. In order to ensure this, the authors have developed a set of use cases for GRCSE. These use cases allowed 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.

Based on the analysis of the GRCSE author team, we have identified five use cases that demonstrate the potential utilization of GRCSE. GRCSE can be used for:

1. Developing SE Courses,
2. Developing SE Curricula,
3. Assessing SE Curriculum,
4. Developing Non-SE Courses (incorporation of SE into courses for other engineering disciplines), and
5. Selecting a Degree.

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. Table 30 represents the relationship between GRCSE users (actors) and use cases (activities); this is further depicted in Figure 13.

Table 30. Use Cases and the Users to Which They Apply. (GRCSE Original)

	SE Faculty	Non-SE Faculty	Professional Trainer	Assessor Evaluator	Student	Employer
SE Course Development	X		X			
SE Curriculum Development	X		X			
SE Curriculum Assessment	X	X	X	X		X
Non-SE Course Dev.	X					
Degree Selection					X	X

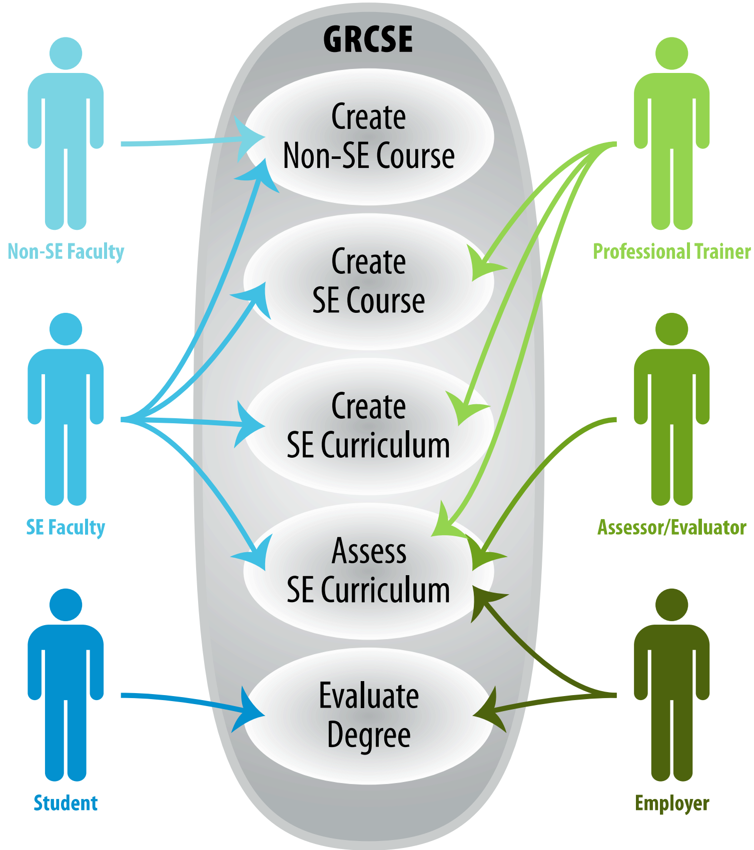


Figure 13. GRCSE Use Case Diagram. (GRCSE Original)

Use Case 1: Systems Engineering Course Development

Goal: A university faculty member or a trainer use GRCSE to design a course on SE.

Actor: A 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 the basic understanding of SE,
5. The user has searched the internet for similar courses,
6. The user has searched for available textbooks on SE,
7. The user has identified an initial set of courses objectives, and
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 breadth and depth of the coverage.

Normal Flow:

1. The user consults GRCSE and reviews:
    a. GRCSE Outcome and Objectives,
    b. GRCSE Core Body of Knowledge (CorBoK), and
    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 SEBOK, for additional information or references that point to additional information; and
8. The user utilizes SEBOK and/or corresponding references to generate course artifacts.

Post-Conditions:

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

Use Case 2: Systems Engineering Curriculum Development

Goal: University faculty member or a trainer uses GRCSE to design the curriculum for graduate program in SE.

Actor: A university faculty member, trainer, group of faculty, and/or academic administrator.

Preconditions and Assumptions:

1. The user has identified the need for the development of new graduate program in SE or the modification of their 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 SEBoK,
5. At least one user has deep knowledge of SE,
6. The user has identified the program constituents, and
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 GRCSE and review:
    a. Entry Requirements,
    b. GRCSE Outcome and Objectives,
    c. GRCSE CorBoK and Extended CorBoK, and
    d. GRSCE CorBoK to SEBoK Relationship;
2. The user establishes an expanded set of outcomes and objectives or expands/modifies 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 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 SEBoK, for additional information or references that point to additional information;
8. The user utilizes SEBoK and/or corresponding references to generate course description and syllabus; and
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 or revised SE graduate curriculum.

Use Case 3: Systems Engineering Curriculum Assessment

Goal: A university faculty member, trainer, assessor/evaluator, and/or employer use GRCSE for the purpose of assessing the curriculum for graduate program in SE.

User: A university faculty member, trainer, assessor/evaluator, employer, etc. (For the purpose of this use case, we refer to this user as an evaluator, but this does not imply they are the official evaluator, e.g. 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 SEBoK,
4. The evaluator has basic knowledge of a SE graduate curriculum, and
5. Outcomes and objectives of the graduate program have been defined and are available for the evaluator.

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

Normal Flow:

1. Evaluator plans the assessment by identifying the goals and objectives;
2. Evaluator performs the assessment:
    a. Evaluator assesses the curriculum outcomes and objectives against GRCSE outcomes and objectives, and
    b. Evaluator review and assess the curriculum content (courses) against CorBoK and extended CorBoK;
3. Evaluator conducts gap analysis based on the results of previous steps; and
4. Evaluator makes a decision based on the result of Gap Analysis.

Variation:

4.a. Evaluator (faculty) decides on:
   i. Whether the gaps between the existing curriculum and GRCSE curriculum should be closed, and
   ii. A way ahead to use GRCSE to modify the curriculum as appropriate (Use Case 1. Systems Engineering Course Development and Use Case 2. Systems Engineering Curriculum Development); and

4.b. Evaluator (employer) decides on whether the graduates of the program would satisfy the required set of knowledge and expertise that the organization needs for its new hires.

Post-Conditions:

1. Gap analysis has been generated, and assessment has been performed, and
2. Changes have been defined and implemented.

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

Goal: A faculty member in an area outside of the SE or a SE faculty use GRCSE to identify SE content that could be included in his/her class(es).

Actor: A non-SE or SE faculty member.

Preconditions and Assumptions:
1. The user has identified a need to include some systems engineering content in his/her class(es),
2. The user has access to GRCSE,
3. The user has access to SEBoK,
4. The user is teaching in engineering field, and
5. The user has some basic understanding of issues in SE.

Trigger: An engineering faculty member has identified a need for inclusion of some SE content as part of his/her class(es).

Normal flow:
1. The user consults GRCSE and reviews:
   a. GRCSE Outcome and Objectives,
   b. GRCSE CorBoK, and
   c. GRSCE CorBoK to SEBoK Relationship;
2. The user establishes a set of SE outcomes and objectives to be included as part of his/her class(es),
3. The user verifies the SE outcomes and objectives against 1.b, and 1.c,
4. The user conducts research in the appropriate/identified areas of systems engineering content that he/she wishes to include in his/her class(es),
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 SEBoK, for additional information or references that point to additional information,
8. The user feels comfortable with the material to be included in his/her class(es), and
9. The user utilizes SEBoK 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: Degree Selection

Goal: A graduate or undergraduate student and/or employer use GRCSE to learn about SE curricula, and/or evaluating graduate programs in SE.

Actor: A graduate or undergraduate student, or employer.

Preconditions and Assumptions:

1. The user has an interest in:
    a. Pursuing graduate study in SE, and/or
    b. Evaluating a SE graduate program offered by a university, or
    c. Learning about graduate SE education; and
2. The user has access to GRCSE.

Trigger: The student has interest in pursuing graduate education in SE, or an employer has an interest in evaluating a graduate SE program.

Normal Flow:

1. The user consults GRCSE and reviews:
    a. GRCSE Outcome and Objectives,
    b. GRCSE Entrance Requirements,
    c. GRCSE CorBoK, and
    d. GRSCE CorBoK to SEBoK Relationship;
2. The user evaluates the content of CorBoK;
3. The user evaluates his/her preparation against entrance requirements;
4. The user confirms his/her interest in pursuing graduate degree in SE;
5. The user identifies potential schools offering a graduate SE degree (Use case 3. Curriculum Assessment);
6. The user compares the candidate schools' curriculum against CorBoK; and
7. The user identifies programs to which he/she would consider applying.

Variation:

3.a. Employer identifies potential schools offering graduate SE degree, and
4.a. Employer identifies strength and weaknesses of the SE degrees against company needs.

Post-Conditions:

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

Appendix H: Example Distribution of Time to Teach The CorBoK

H.1 Background

Chapter 6 describes the purpose, nature, organization, and rationale for the CorBoK, which is based on the SEBoK. For purposes of planning curriculum content, GRCSE estimates 480 hours as the total number of contact hours needed for a SE master's program. Per the guidance in Chapter 6, CorBoK instruction should take approximately 50% of the time spent on a master's program; i.e., 240 contact hours. Clearly, this will vary from program to program, but the general guidance of 240 contact hours for coverage of CorBoK topics provides a key element in GRCSE curriculum design.

It is assumed that, in general, topics at higher Bloom's levels require more contact hours. For example, Part 2 has 25 topics, almost all at the Comprehension level, and Part 3 has 34 topics, distributed over the Comprehension, Application, and Analysis levels. Hence, Part 3 would require more contact hours than Part 2. The recommended allocation is approximately 30% of the time should be devoted to the Part 3 CorBoK topics and approximately 20% of the time to topics in Parts 2, 4, 5, and 6 in total.

H.2 Example Distribution of Time Devoted to the CorBoK

Table 31 provides an example of how the 240 contact hours could be distributed across the CorBoK parts. Contact hours (rounded to the nearest whole) and percentages of the 50% devoted to CorBoK are provided for each part. Since Part 3 represents a significant portion of the 50%, distribution by KA is presented. The example is meant to be typical, but many other distributions are clearly possible. The example distribution was determined by engaging a group of GRCSE authors in a quasi-Wideband Delphi technique to allocate the 240 contact hours, and then the hours were converted to percentages.

Table 31. Example Distribution of Time to The CorBoK. (GRCSE Original)

CorBoK Part	Foundation/SEM Contact Hours (%)	Foundation/SDD Contact Hours (%)
Part 2: Systems	29 (6%)	29 (6%)
Part 3: Systems Engineering and Management	134 (28%)	134 (28%)
Life Cycle Models	19 (4%)	19 (4%)
System Definition	19 (4%)	29 (6%)
System Realization	19 (4%)	29 (6%)
System Deployment and Use	19 (4%)	19 (4%)
Systems Engineering Management	24 (5%)	10 (2%)
Product and Service Life Management	24 (5%)	19 (4%)
Systems Engineering Standards	10 (2%)	10 (2%)
Part 4: Applications of Systems Engineering	24 (5%)	24 (5%)
Part 5: Enabling Systems Engineering	24(5%)	10 (2%)
Part 6: Related Disciplines	29 (6%)	43 (9%)
Total Distribution - Contact Hours (%)	**240 (50%)**	**240 (50%)**

The contact hours on KAs can be packaged into courses in many ways by a program. For example, Table 31 shows 28% or 134 contact hours being devoted to Systems Engineering and Management topics.

Using a typical US three credit hour course, this would equate to the equivalent of about three courses. But these contact hours could be spent across several courses, including the Capstone Experience, along with topics from the other CorBoK parts or other program specialties.

As an example, a course in Systems Requirements Analysis would certainly draw content from Part 3 KA on System Definition, but it might also cover material from Part 2 (e.g., on topics in Systems Fundamentals, Systems Thinking, and Systems Modeling) and Part 4 (e.g., on topics in Product SE and Services SE). In addition, some of the application and analysis levels of capability in the Systems Definition KA might not be reached in such a course, but might only be accomplished in a subsequent course (e.g. a course focused on the Capstone Experience).

SE faculty could start with Table 31 and revise the hours and percentages to meet their program objectives, outcomes, and delivery mode. Then the revised table could be used to either assess an existing curriculum or as a driving factor in designing a new curriculum.