BKCASE: *Guide to The Body of Knowledge and Curriculum to Advance Systems Engineering*

Arthur Pyster
Stevens Institute of Technology

BKCASE is pronounced "bookcase"

APCOSE 2010
Systems Engineering: Collaboration for Intelligent Systems

What is BKCASE?

Project to create:

- Systems Engineering Body of Knowledge
- Graduate Reference Curriculum in Systems Engineering (GRCSE™ – pronounced “Gracie”)

Started in September 2009 by Stevens Institute of Technology and Naval Postgraduate School with primary support from Department of Defense

Project will run through 2012

Intended for world-wide use

What is the SEBoK?

Describes the boundaries, terminology, content, and structure of SE that are needed to systematically and consistently *support:*

Task Name	Task Description
Inform Practice	Inform systems engineers about the boundaries, terminology, and structure of their discipline and point them to useful information needed to practice SE in any application domain
Inform Research	Inform researchers about the limitations and gaps in current SE knowledge that should help guide their research agenda
Define Curricula	Define the content that should be common in undergraduate and graduate programs in SE
Certify Professionals	Certify individuals as qualified to practice systems engineering
Decide Competencies	Decide which competencies practicing systems engineers should possess in various roles ranging from apprentice to expert

Guide to the literature, not all the content of the literature

What is in GRCSE?

Guidance for Constructing and Maintaining the Reference Curriculum: the fundamental principles, assumptions, and context for the reference curriculum authors

Entrance Expectations: what students should be capable of and have experienced before they enter a graduate program

Outcomes: what students should achieve by graduation

Architecture: the structure of a curriculum to accommodate core material, university-specific material, and elective material

Core Body of Knowledge: material that all students should master in a graduate SE program

Not specific courses. Not specific packaging. Adaption and selective adoption expected and encouraged.

BKCASE Vision and Objectives

Vision

"Systems Engineering competency models, certification programs, textbooks, graduate programs, and related workforce development initiatives around the world align with BKCASE."

Objectives

1. Create the SEBoK and have it be globally recognized by the SE community as the authoritative guide to the body of knowledge for the SE discipline.
2. Create GRCSE and have it be globally recognized by the SE community as the authoritative guidance for graduate programs in SE.
3. Facilitate the global alignment of related workforce development initiatives with SEBoK and GRCSE.
4. Transfer stewardship of SEBoK and GRCSE to INCOSE and the IEEE after BKCASE publishes version 1.0 of those products, including possible integration into their certification, accreditation, and other workforce development and education initiatives.

Our Partners

Under consideration

Under consideration

Author	Organization
Rick Adcock	Cranfield University, UK
Johann Amsenga	Eclipse RDC, South Africa
Erik Aslaksen	Sinclair Knight Merz, Australia
John Baras	University of Maryland, US
Barry Boehm	University of Southern California, US
John Brackett	Boston University, US
Aaron Eng Seng Chia	National University of Singapore, Singapore
Edmund Conrow	Management and Technology Associates, US
Paul Croll	Computer Sciences Corporation, US
Cihan Dagli	Missouri University of Science and Technology, US
Heidi Davidz	UTC Pratt & Whitney, US
Joseph J. Ekstrom	Brigham Young University, US
Marcia Enos	Lockheed Martin, US
Dick Fairley	IEEE, US
Alain Faisandier	Association Francaise d'lingeniere Systeme, France
Tim Ferris	University of South Australia, Australia
Kevin Forsberg	Center for Systems Management, US
G. Richard Freeman	Air Force Institute of Technology, US
Sanford Friedenthal	Lockheed Martin, US

Authors as of September 2010

Author	Organization
Richard Frost	General Motors, US
Brian Gallagher	Northrop Grumman, US
Edward Ghafari	ICES in US
Tom Hilburn	Embry-Ridlle Aeronautical University, US
Nicole Hutchison	Stevens Institute of Technology, US
Scott Jackson	University of Southern California, US
Ken Kepchar	Federal Aviation Administration, US
Naohiko Kohtake	Keio University , Japan
Mike Krueger	ASE Consulting, Australia
Harold “Bud” Lawson	Lawson Konsult AB, Sweden
Yeaw lip “Alex” Lee	Defence Science and Technology Agency, Singapore
Ray Madachy	Naval Postgraduate School, US
James Martin	Aerospace Corporation, US
Greg Mayhew	The Boeing Company, US
Andrew McGettrick	Association for Computing Machinery, UK
Ken Nidiffer	Software Engineering Institute, US
Dave Olwell	Naval Postgraduate School, US

BKCASE™

Authors, *cont.*
as of
September 2010

Author	Organization
Daniel Prun	Ecole Nationale de l'Aviation Civile (ENAC), France
Art Pyster	Stevens Institute of Technology, US
Garry Roedler	Lockheed Martin, US
Jean-Claude Roussel	EADS, France
Sven-Olaf Schulze	Berner & Mattner, Germany
Seiko Shiraska	KEIO University, Japan
Hillary Sillitto	Thales Group, UK
John Snoderly	Defense Acquisition University, US
Alice Squires	Stevens Institute of Technology, US
Massood Towhidnejad	Embry-Riddle Aeronautical University, US
Guilherme Horta Travassos	Federal University of Rio de Janeiro, Brazil
Mary VanLeer	Arkansas Scholarship Lottery, US
Qing Wang	Institute of Software Chinese Academy of Sciences, China
Brian Wells	Raytheon, US

Authors, *cont.*
as of
September 2010

Rules for BKCASE Activities

1. Products generated by the authors, not the sponsor or partners
2. Even though the Department of Defense is the sponsor, it does not have any authority over the content of the products, nor are the products slanted towards defense systems development and acquisition
3. Volunteer authors do the bulk of the writing
4. Core Team from Stevens and Naval Postgraduate School provides stable labor and direction
5. Core Team responsible for final integration, technical editing, and clean up of products

How We Got Here

In Spring 2007, 3 phase effort was proposed:

1. A reference curriculum* for graduate software engineering with the “right” amount of systems engineering
2. A reference curriculum for graduate systems engineering with the “right” amount of software engineering
3. A fully interdisciplinary reference curriculum for systems and software engineering

*A reference curriculum offers recommendations on outcomes at graduation, entrance expectations, curriculum architecture, required knowledge, and possibly objectives. Recommendations are expected to be tailored. They do not include specific course descriptions or other packaging.

Phase 1 Primary Products

Graduate Software Engineering 2009 (GSwE2009): Curriculum Guidelines for Graduate Degree Programs in Software Engineering

GSwE2009 Companion Document: Comparisons of GSwE2009 to Current Master's Programs in Software Engineering

GSwE2009 Companion Document: Frequently Asked Questions on Implementing GSwE2009


iSSEc

Integrated Software & Systems Engineering Curriculum (iSSEc) Project

Graduate Software Engineering 2009(GSwE2009)

Curriculum Guidelines for
Graduate Degree Programs in Software Engineering

GSwE 2009

Version 1.0

September 30, 2009

Unlimited Global Distribution

© 2009 Stevens Institute of Technology

i

Endorsed by INCOSE, NDIA SE Division, Brazilian Computer Society

Originally sponsored by DoD. Now sponsored by the IEEE Computer Society and ACM

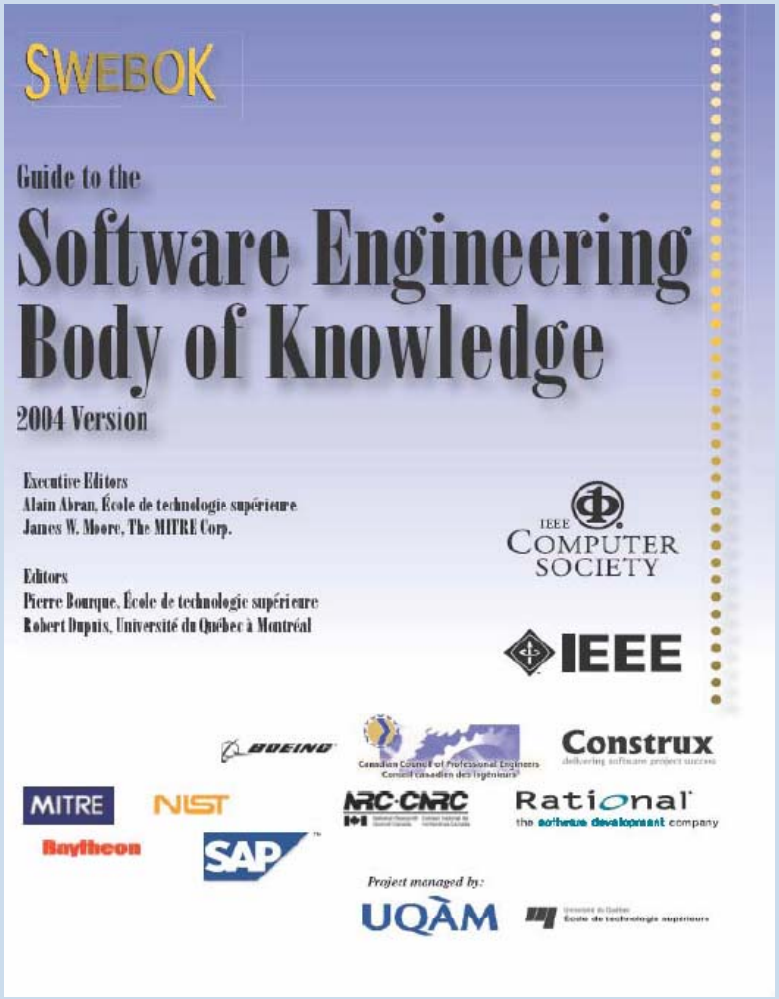
www.GSwE2009.org

SEBoK Value Proposition

1. There is no authoritative source that defines and organizes the knowledge of the SE discipline. Knowledge gap creates unnecessary inconsistency and confusion in understanding the role of SE and in defining SE products and processes.
2. Creating the SEBoK will help build community consensus on the boundaries of SE, including its entanglements with project management and software engineering.
3. A common way to refer to SE knowledge will facilitate communication among systems engineers and provide a baseline for competency models, certification programs, educational programs, and other workforce development initiatives around the world.
4. Common ways to identify metadata about SE knowledge will facilitate search and other automated actions on SE knowledge.

What Has *Software Engineering* Done to Address Similar Challenges?

SWEBOK


SWEBOK
Guide to the
Software Engineering
Body of Knowledge
2004 Version
Executive Editors
Alain Abran, École de technologie supérieure
James W. Moore, The MITRE Corp.
Editors
Pierre Bourque, École de technologie supérieure
Robert Dupuis, Université du Québec à Montréal
IEEE COMPUTER SOCIETY
IEEE
BOEING
Construx
MITRE
NIST
NRC·CNRC
Rational
the software development company
Raytheon
SAP
Project managed by:
UQÀM

SWEBOK

SWEBOK is a way of organizing all the knowledge that is within the software engineering (SwE) discipline

It is a hierarchical structure for the knowledge and references to key documents stating the knowledge as of 2004

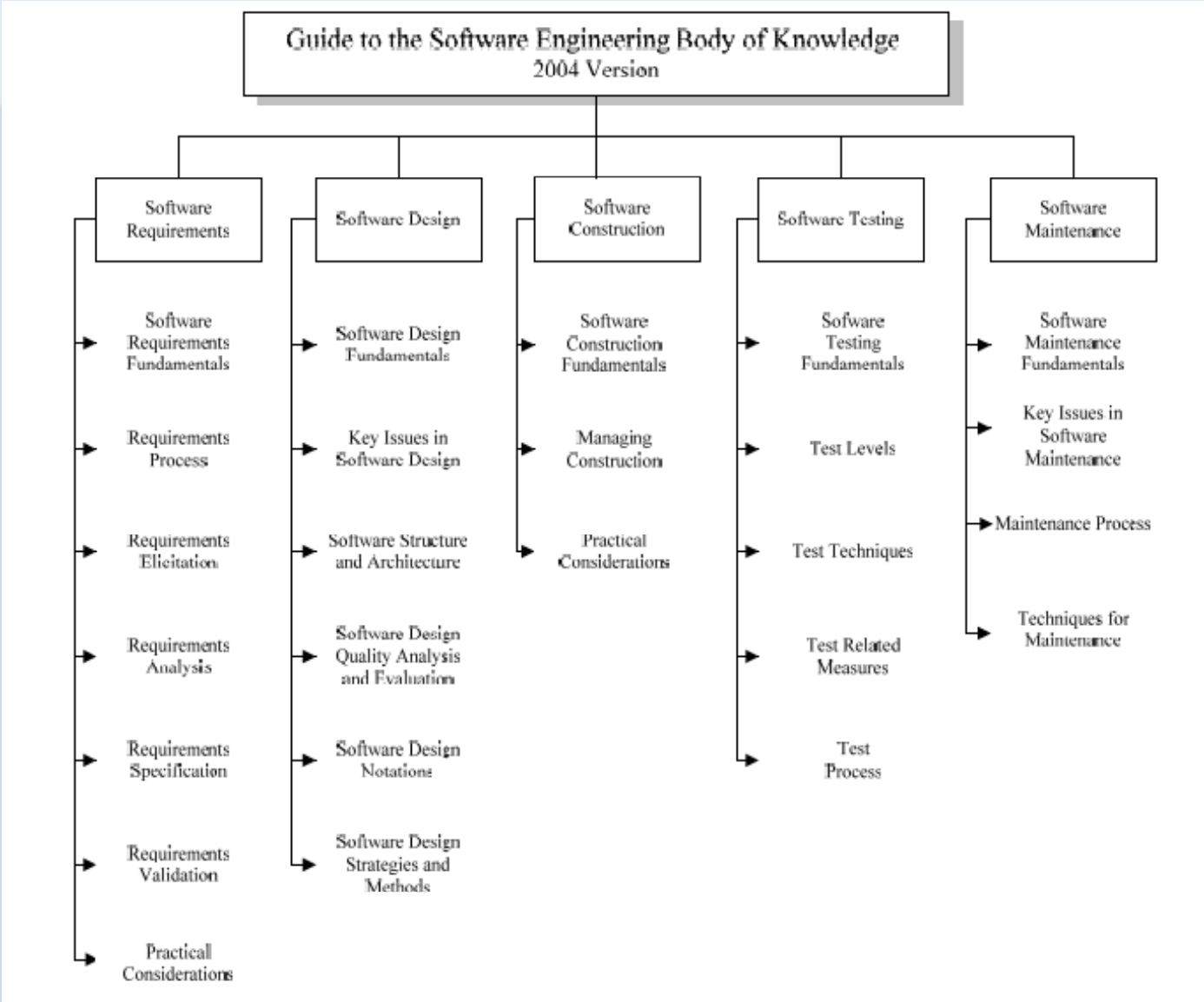
It was developed by a community of authors and reviewers from around the world

It is static – it has not changed since it was published

A refresh project is underway to produce a new version in 2010

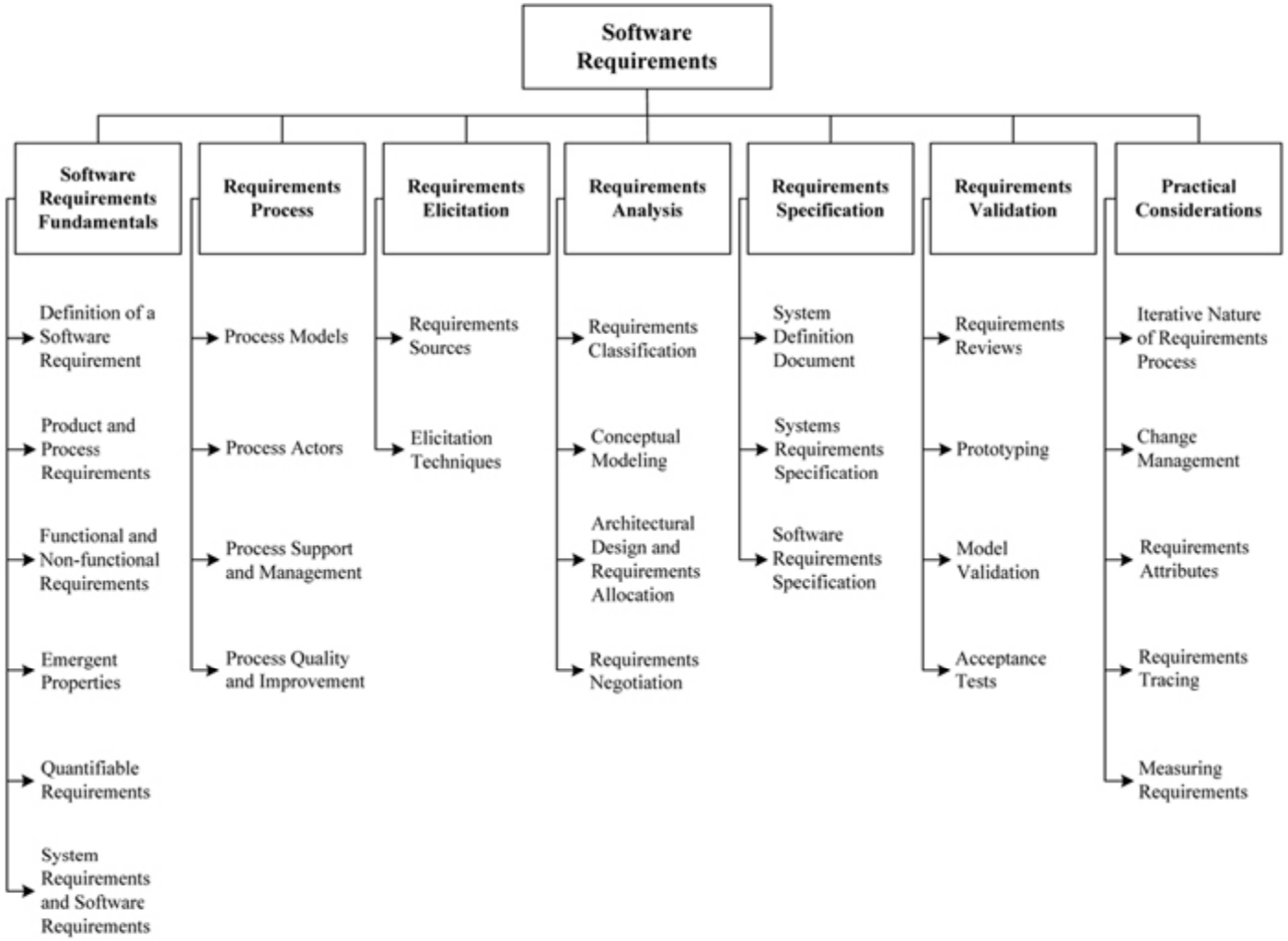
www.SWEBOK.org

11 Knowledge Areas, First 5



SWEBOK Example

BREAKDOWN OF TOPICS FOR SOFTWARE



SWEBOK Topic Text

1. Software Requirements Fundamentals

1.1. Definition of a Software Requirement

At its most basic, a software requirement is a property which must be exhibited in order to solve some problem in the real world. The Guide refers to requirements on "software" because it is concerned with problems to be addressed by software. Hence, a software requirement is a property which must be exhibited by software developed or adapted to solve a particular problem. The problem may be to automate part of a task of someone who will use the software, to support the business processes of the organization that has commissioned the software, to correct shortcomings of existing software, to control a device, and many more. The functioning of users, business processes, and devices is typically complex. By extension, therefore, the requirements on particular software are typically a complex combination of requirements from different people at different levels of an organization and from the environment in which the software will operate.

An essential property of all software requirements is that they be verifiable. It may be difficult or costly to verify certain software requirements. For example, verification of the throughput requirement on the call center may necessitate the development of simulation software. Both the software requirements and software quality personnel must ensure that the requirements can be verified within the available resource constraints.

Requirements have other attributes in addition to the behavioral properties that they express. Common examples include a priority rating to enable trade-offs in the face of finite resources and a status value to enable project progress to be monitored. Typically, software requirements are uniquely identified so that they can be over the entire software life cycle. [Kot00; Pfl01; Som05; Tha97]

1.2. Product and Process Requirements ▲

A distinction can be drawn between product parameters and process parameters. Product parameters are requirements on software to be developed (for example, "The software shall verify that a student meets all prerequisites before he or she registers for a course.").

A process parameter is essentially a constraint on the development of the software (for example, "The software shall be written in Ada."). These are sometimes known as process requirements.

Some software requirements generate implicit process requirements. The choice of verification technique is one example. Another might be the use of particularly rigorous analysis techniques (such as formal specification methods) to reduce faults which can lead to inadequate reliability. Process requirements may also be imposed directly by the development organization, their customer, or a third party such as a safety regulator [Kot00; Som97].

SEBoK Content

1. The definition of fundamental terms and concepts and primary relationships between those concepts
2. A statement of the principles of SE
3. A description of generally accepted activities, practices, technologies, processes, methods, and artifacts of SE and how they relate to one another
4. How the knowledge of SE varies within individual application domains such as medicine, transportation, and telecommunications
5. References to books, articles, websites, and other sources that elaborate on the information in the SEBoK

SEBoK 0.25 Table of Contents

1. Introduction
2. System Concepts and Systems Thinking
3. SE Overview
4. Generic Life Cycle Stages
5. Service SE
6. Enterprise SE
7. Enabling SE in the Organization
8. SE Management
9. System Definition
10. System Realization Fundamentals
11. System Deployment and Use
12. System Life Management
13. SE Agreement
14. Cross-Cutting Knowledge
15. SE Competencies
16. SE Applications/Case Studies
17. References
18. Glossary

SEBoK Status

1. Version 0.25 released for limited review on September 15, 2010
2. 656 pages long, 15 out of 16 chapters drafted, lots of very good material, but uneven maturity, too aerospace/defense, too process-oriented
3. Around 200 reviewers signed up – hoping for > 1000 comments – direction, topics, style, references, …
4. Reviews due December 15, 2010
5. All review comments will be adjudicated and adjudication published
6. Case studies planned for early 2011
7. Version 0.5 (September 2011)will be hyperlinked and may have wiki-style governance

GRCSE Value Proposition

1. There is no authoritative source to guide universities in establishing the outcomes graduating students should achieve with a master's degree in SE, nor guidance on reasonable entrance expectations, curriculum architecture, or curriculum content.

2. This gap in guidance creates unnecessary inconsistency in student proficiency at graduation, makes it harder for students to select where to attend, and makes it harder for employers to evaluate prospective new graduates.

GRCSE is being created analogously to GSwE2009 – in fact, using GSwE2009 as the starting text

Version 0.25 expected in December 2010

GRCSE 0.25 Draft Contents

Title - Chapters
1. Introduction
2. Guidance for the construction and maintenance of GRCSE
3. Expected objectives when a graduate has 3-5 years' experience
4. Expected outcomes when a student graduates
5. Expected student background when entering master's program
6. Curriculum architecture
7. Core body of knowledge (CorBOK)
8. Assessment
9. Anticipated GRCSE evolution

Title - Appendices
App A. Summary of Graduate SE-centric SE programs in 2010
App B. Bloom's taxonomy of educational objectives
App C. Systems engineering competency frameworks
References
Glossary
Index

Early Decisions (subject to change)

1. Expect an undergraduate degree in engineering, physical science, or mathematics.
2. At least two years of practical experience in some aspect of systems engineering before entering program.
3. No more than 50% of content should be required in recommendations.
4. Accommodate a range of program focuses from more technical to more managerial
5. Learn systems engineering in the context of an application domain

If We Are Successful…

SEBoK will strongly influence the INCOSE SE Handbook Version 4, the INCOSE SE Professional Certification Program, DoD SE competency efforts, will highlight places where research is needed, become a standard reference for practitioners, and improve the quality and richness of communication among systems engineers worldwide.

GRCSE will clearly distinguish between graduate and undergraduate education in SE and influence the content of both undergraduate and graduate SE programs worldwide.

Interested in helping? We need reviewers, subject matter experts, and a few more authors

www.BKCASE.org

bkcase@stevens.edu