Raytheon

Customer Success Is Our Mission

GSwE2009 Graduate Software Engineering Curriculum Model

Dennis J. Frailey

Principal Fellow, Raytheon

Adjunct Professor, Southern Methodist Univ.

DJFrailey@Raytheon.com

Frailey@lyle.smu.edu

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE **APR 2010**	2. REPORT TYPE	3. DATES COVERED **00-00-2010 to 00-00-2010**

4. TITLE AND SUBTITLE **GSwE2009 Graduate Software Engineering Curriculum Model**	5a. CONTRACT NUMBER
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Raytheon,Network Centric Systems,2501 West University Drive,McKinney,TX,75071**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR’S ACRONYM(S)
	11. SPONSOR/MONITOR’S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

Presented at the 22nd Systems and Software Technology Conference (SSTC), 26-29 April 2010, Salt Lake City, UT. Sponsored in part by the USAF. U.S. Government or Federal Rights License

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**43**	

History of the Project

- Initiated in 2007 by Kristen Baldwin, then Deputy Director for **Software Engineering and System Assurance** of the Office of the Under Secretary of Defense Acquisition, Technology and Logistics
- Baldwin approached Art Pyster of Stevens Institute regarding the findings of a **software industrial base study** that had been conducted at the request of Office of the Secretary of Defense.
- Baldwin and Pyster concluded that a **critical long-term strategy** for the Department was to **ensure a strong and relevant foundation for training and education of senior software and systems talent**
- The approach selected was to establish a **reference curriculum for each discipline** that would represent the fundamentals as well as address the current challenges of scale, complexity, and criticality.

Some of the Key Findings

- One of the major causes of defense program problems was found to be **poor communication between systems and software engineers**
  - Different organizational entities within most defense contractors
  - Different languages, processes, models, methods, tools
  - Different academic backgrounds
  - Etc. etc.
- Furthermore, few engineers of either variety are familiar with the **latest developments in their own fields**
  - Too busy working on their programs to keep up with rapidly advancing disciplines
  - Often, each company has its own unique culture for each of these disciplines that does not necessarily align with what is done outside
- And, finally, the **existing academic programs in these fields** are so **inconsistent** that contractors and the government do not always know what they are getting when they hire someone with a graduate degree in either field.

What the DoD Needs

1. Better coordination between these fields in defense contractor organizations
2. “Raising the bar” on the content of graduate programs in these fields
3. More consistency among graduate programs in these fields

The ISSEC project is intended to address the second and third items in the above list.

The Integrated Software and Systems Engineering Curriculum Project

- Begun in May 2007 at Stevens Institute of Technology
- Sponsored by DoD Director of Systems and Software Engineering (Office of the Secretary of Defense)
- Four products planned:

  1. A modern reference curriculum for a master's degree in <u>software engineering</u> that integrates an appropriate amount of systems engineering
  2. A modern body of knowledge for <u>systems engineering</u>
  3. A modern reference curriculum for a master's degree in <u>systems engineering</u> that integrates an appropriate amount of software engineering
  4. A truly <u>interdisciplinary degree</u> that is neither systems nor software engineering – it is both

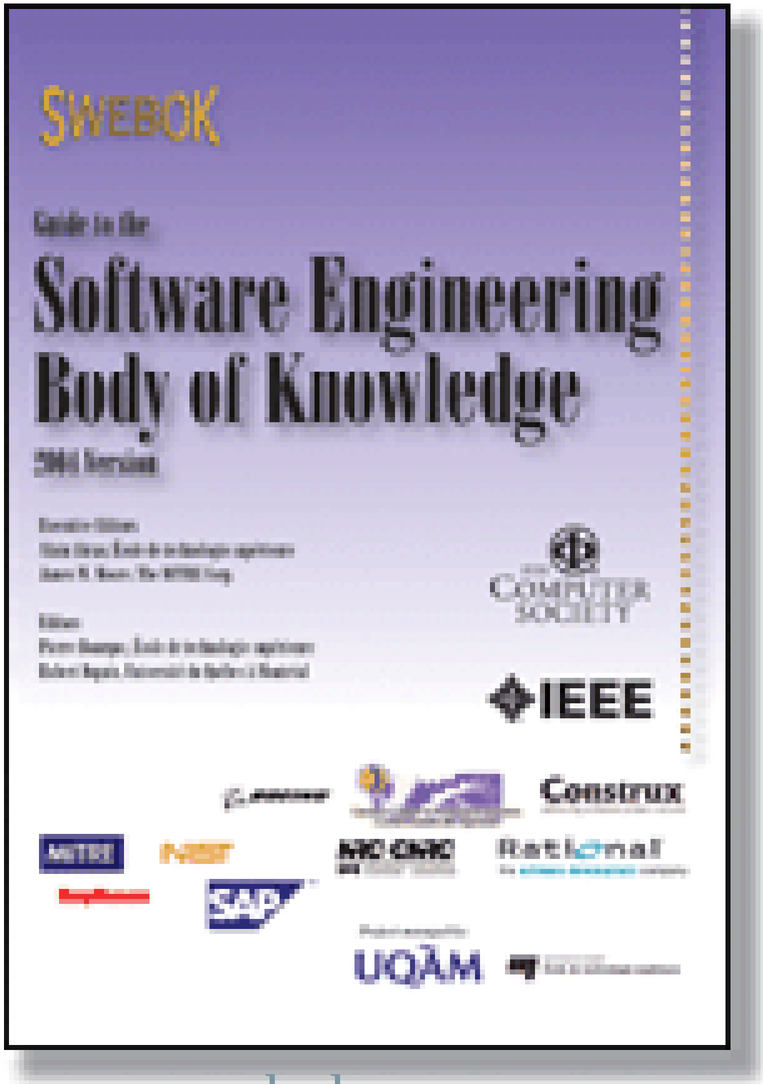
What do we Teach for a Master’s Degree in Software Engineering?

- The last effort to create a reference curriculum for graduate software engineering education was by the Software Engineering Institute (SEI) in the **early 1990s**.
- There are, in effect, **no current community-endorsed recommendations on what to teach software engineers at the graduate level** – nothing that recognizes how the world has changed.
- **Response**: create a project to create a new reference curriculum in software engineering

1st Project – Graduate Software Engineering Reference Curriculum

1. Understand the current state of SwE graduate education (November 2007)
2. Create GSwERC 0.25 with a small team, suitable for limited review (February 2008)
3. Publicize effort through conferences, papers, website, etc (continuous)
4. Create GSwERC 0.50 suitable for broad community review and early adoption (October 2008)
5. Create GSwE2009 suitable for broad adoption (2009)
6. Transition stewardship to professional societies (2009-2010)
7. Foster adoption world-wide (2010 and beyond)

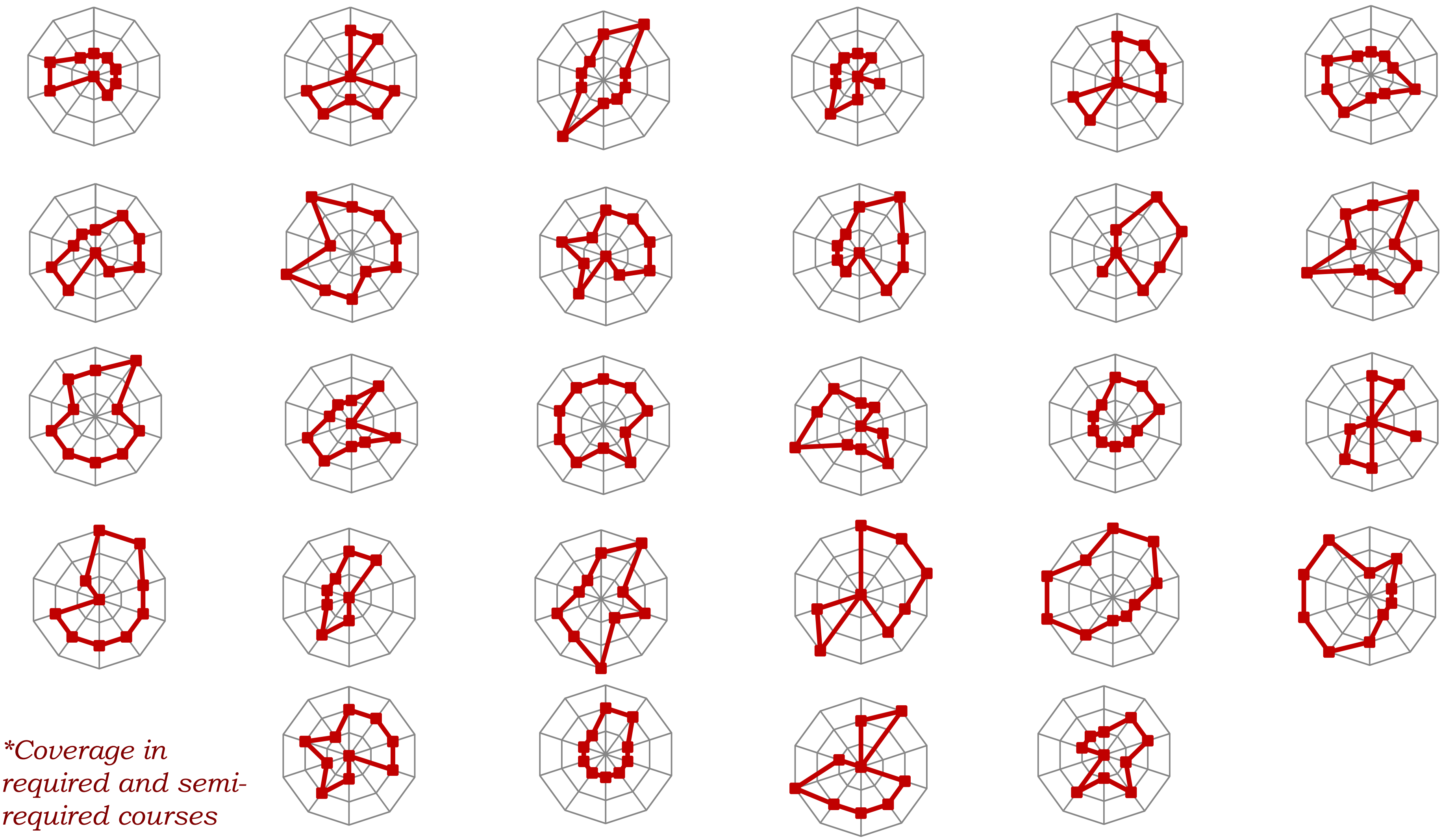
Body of Knowledge - SWEBOK



www.swebok.org

- Software Requirements Analysis
- Software Design
- Software Construction
- Software Testing
- Software Maintenance
- Software Configuration Management
- Software Engineering Management
- Software Engineering Process
- Software Engineering Tools and Methods
- Software Quality

SWEBOK coverage* in 2007 across 28 SwE MS programs



**Coverage in required and semi-required courses*

The Curriculum Author Team

- Rick Adcock, *Cranfield University and INCOSE participant*
- Mark Ardis, *Rochester Institute of Technology*
- Larry Bernstein, *Stevens Institute of Technology*
- Barry Boehm, *University of Southern California*
- Pierre Bourque, *École de technologie supérieure and SWEBOK volunteer*
- John Bracket, *Boston University*
- Murray Cantor, *IBM*
- Lillian Cassel, *Villanova and ACM participant*
- Robert Edson, *ANSER*
- Richard Fairley, *Colorado Technical University*
- **Dennis Frailey, Raytheon** *& Southern Methodist University*
- Gary Hafen, *Lockheed Martin and NDIA participant*
- Thomas Hilburn, *Embry-Riddle Aeronautical University*
- Greg Hislop, *Drexel University and IEEE Computer Society participant*
- Dave Klappholz, *Stevens Institute of Technology*
- Philippe Kruchten, *University of British Columbia*
- Phil Laplante, *Pennsylvania State University, Great Valley*
- Scott Lucero, *Department of Defense*
- Qiaoyun (Liz) Li, *Wuhan University, China*
- James McDonald, *Monmouth University*
- John McDermid, *University of York, UK*
- Ernest McDuffie, *National Coordination Office for NITRD*
- Bret Michael, *Naval Postgraduate School*
- Ken Nidiffer, *Software Engineering Institute*
- Art Pyster, *Stevens Institute of Technology*
- Mary Shaw, *Carnegie Mellon University*
- Robert Suritis, *IBM*
- Richard Thayer, *California State University at Sacramento*
- Barrie Thompson, *Sunderland University, UK*
- Guilherme Travassos, *Brazilian Computer Society, Brazil*
- Richard Turner, *Stevens Institute of Technology*
- Joseph Urban, *Texas Technical University*
- Ricardo Valerdi, *MIT & INCOSE participant*
- David Weiss, *Avaya*
- Mary Jane Willshire, *Colorado Technical University*

Phase 1 Primary Products

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

- GSwE2009 Companion Document: **Comparisons of GSwE2009 with Current Master's Programs** in Software Engineering

- GSwE2009 Companion Document: Frequently Asked Questions on **Implementing GSwE2009**

www.GSwE2009.org

iSSEc

Integrated Software & Systems Engineering Curriculum (iSSEc) Project

Graduate Software Engineering 2009(GSwE2009)

Curriculum Guidelines for
Graduate Degree Programs in Software Engineering

Version 1.0

September 30, 2009

Unlimited Global Distribution

© 2009 Stevens Institute of Technology

i

Endorsed by INCOSE, NDIA SE Division, Brazilian Computer Society

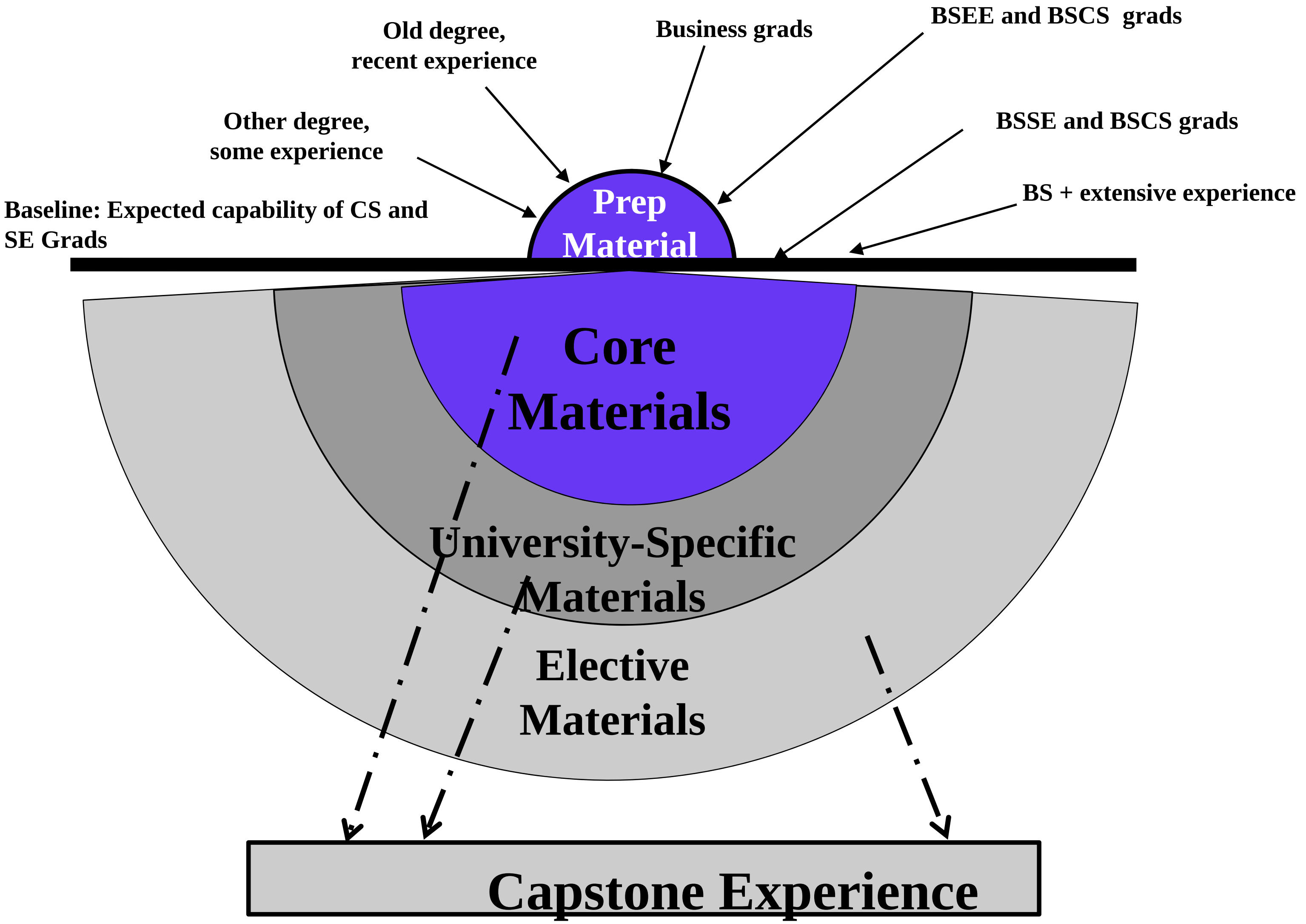
Sponsored by DoD, IEEE Computer Society and ACM

Transition to Bret[redacted]chael
– The Structure of GSwE2009

How the Curriculum is Organized

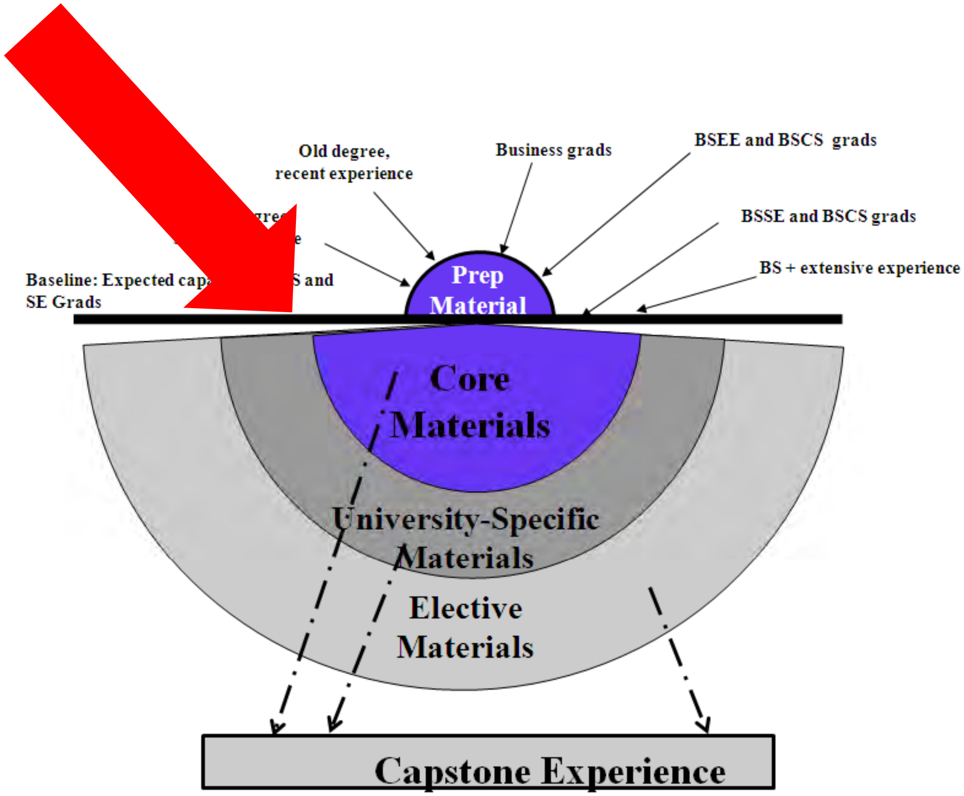
- What they Know when they Start
- What they Learn
- What they are Expected to Achieve (outcomes)

Structure of GSwE2009 Model

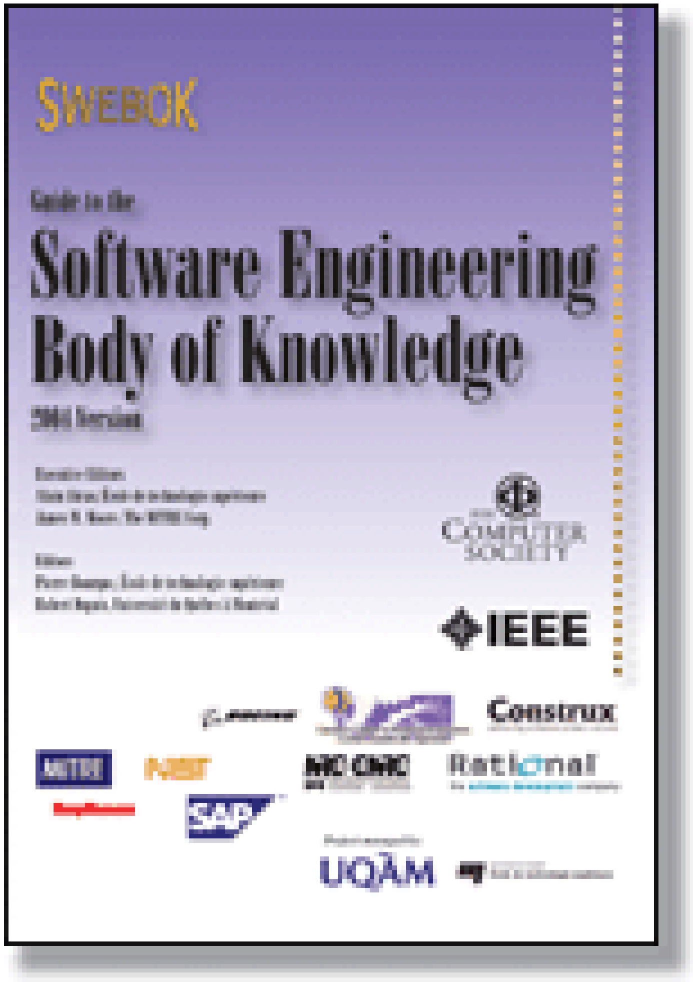


Entry Level Expectations

- The equivalent of an undergraduate degree in computing or an engineering or scientific field with a minor in computing.
- The equivalent of an introductory course in software engineering.
- At least two years of practical experience in some aspect of software engineering or software development.



Core Material is Based on SWEBOK



SWEBOK

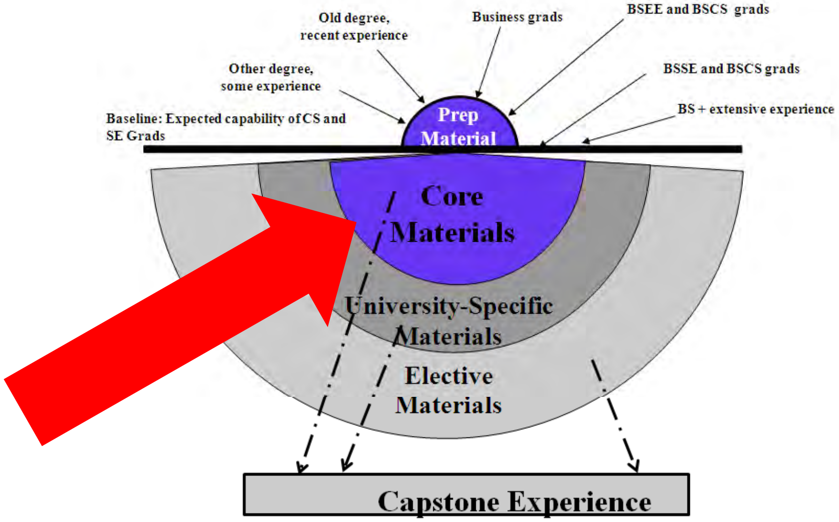
Guide to the

Software Engineering Body of Knowledge

2004 Version

Bloom's Knowledge Levels are Used to Define the Depth of Knowledge

- Knowledge (K)
- Comprehension (C)
- Application (AP)
- Analysis (AN)
- Synthesis (SY)



Core Body of Knowledge
Math Fundamentals

Knowledge Areas	Bloom's Level
Mathematics Fundamentals	
1. **Discrete Structures** Functions, relations, and sets; basic logic; proof techniques; basics of counting; graphs and trees; discrete probability	**AP**
2. **Propositional and Predicate Logic** Propositions, operators and truth tables, laws of logic, predicates and quantifiers, argument and inference.	**AP**
3. **Probability and Statistics** Basic probability theory, random variables and probability distributions, estimation theory, hypothesis testing, regression analysis, analysis of variance.	**AP**

Core Body of Knowledge
Computing Fundamentals

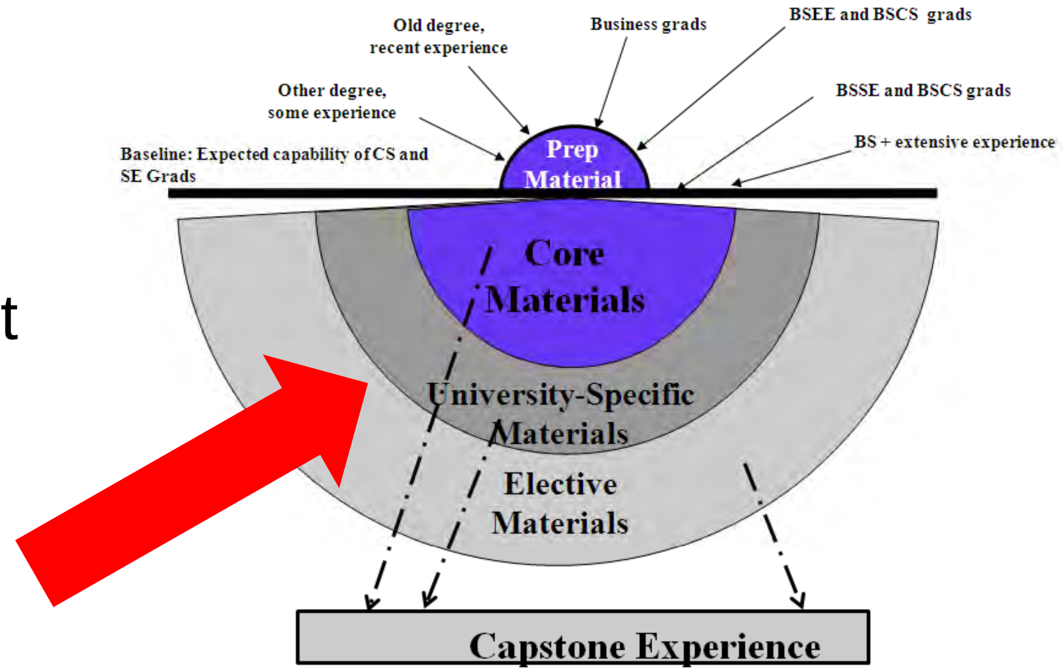
Computing Fundamentals	
1. **Programming Fundamentals**	**AP**
Overview of programming languages; virtual machines; introduction to language translation; declaration and types; abstraction mechanisms; object-oriented programming; functional programming; language translation systems; type systems; programming language semantics; programming language design	
2**. Data Structures and Algorithms**	**C**
Basic algorithmic analysis; algorithmic strategies; fundamentals of computing algorithms; distributed algorithms	
3. **Computer Architecture**	**C**
Digital logic and digital systems; machine level representation of data; assembly level machine organization; memory system organization and architecture; interfacing and communication; functional organization; multiprocessing and alternative architectures; performance enhancements; architecture for networks and distributed systems	
4. **Operating Systems**	**C**
Operating system overview & principles; concurrency; scheduling and dispatch; memory management; device management; security and protection; file systems; real-time and embedded systems; fault tolerance; system performance evaluation; scripting	
5. **Networks and Communications**	**C**
Introduction to net-centric computing; communication and networking; network security; Internet; building web applications; network management; compression and decompression; multimedia data technologies; wireless and mobile computing	
6. **Module Design and Construction**	**AP**

Core Body of Knowledge
Software Engineering Fundamentals

Software Engineering	
1. **Software Requirements**	C
Software requirements fundamentals; requirements elicitation; requirements analysis; requirements specification; requirements validation	
2. **Software Design**	C
Software design fundamentals; software structure and architecture; software design notations; software design strategies and methods	
3. **Software Construction**	AP
Software construction fundamentals; software construction practices	
4. **Software Testing**	K
Software testing fundamentals; test levels; test techniques	
5. **Software Maintenance**	K
Software maintenance fundamentals; techniques for maintenance	
6. **Software Engineering Management**	K
Software project planning; software configuration management	
7. **Software Engineering Process**	K
Process definition and implementation; product and process measurement	
8. **Software Quality**	K

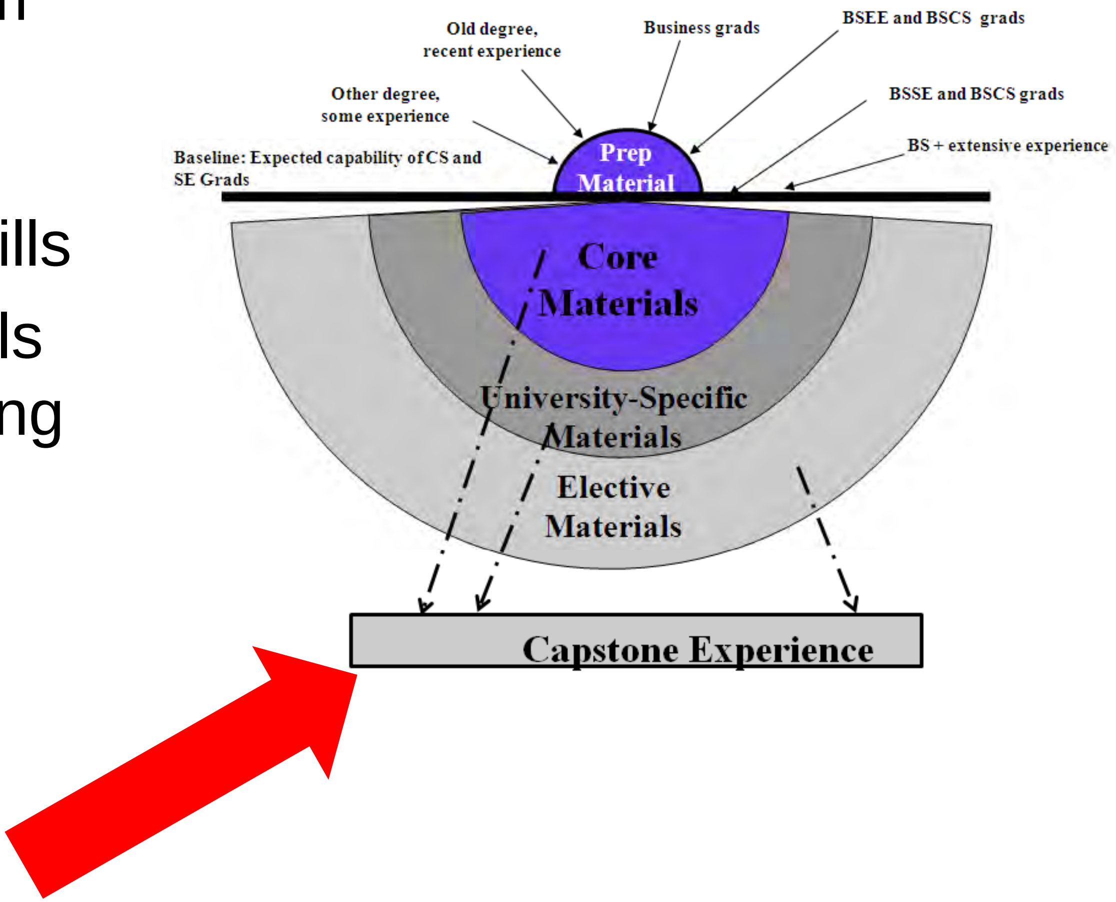
University and Elective Material are Unique to the Institution & Specialty

- They must achieve depth in at least one application domain
- They must achieve synthesis level (highest Bloom level) in at least one of the core knowledge areas



Capstone Experience

- A project, a practicum or a thesis
- That demonstrates their accumulated skills
- Integrating those skills in the course of solving a problem



Outcomes

- This is how we evaluate the results of the curriculum
- Ten outcomes were identified based on the overall goals of the curriculum

Outcomes: A student completing the curriculum will …

1. Have mastered the Core Body of Knowledge (CBOK).
2. Have mastered at least one application domain.... That mastery includes understanding how differences in domain and type manifest themselves in both the software itself and in their engineering, and includes understanding how to learn a new application domain or type.
3. Have mastered at least one knowledge area or sub-area from the CBOK to at least the Bloom Synthesis level.
4. Have demonstrated how to make ethical professional decisions and practice ethical professional behavior.
5. Understand the relationship between software engineering and systems engineering and be able to apply systems engineering principles and practices in the engineering of software.

Outcomes: A student completing the curriculum will …

6. Be able to work effectively as part of a team, including teams that may be international and geographically distributed, to develop quality software artifacts, …
7. Be able to reconcile conflicting project objectives, finding acceptable compromises within limitations of cost, time, knowledge, existing systems, and organizations.
8. Understand and appreciate the importance of feasibility analysis, negotiation, effective work habits, leadership, and good communication with stakeholders …
9. Understand how to learn new models, techniques, and technologies as they emerge, and appreciate the necessity of such continuing professional development.
10. Be able to analyze a current significant software technology, articulate its strengths and weaknesses, and specify and promote improvements or extensions to that technology.

Raytheon
Customer Success Is Our Mission

Comparisons of GSwE2009 with 12 Existing Programs

Dennis J. Frailey

Principal Fellow, Raytheon

Adjunct Professor, Southern Methodist Univ.

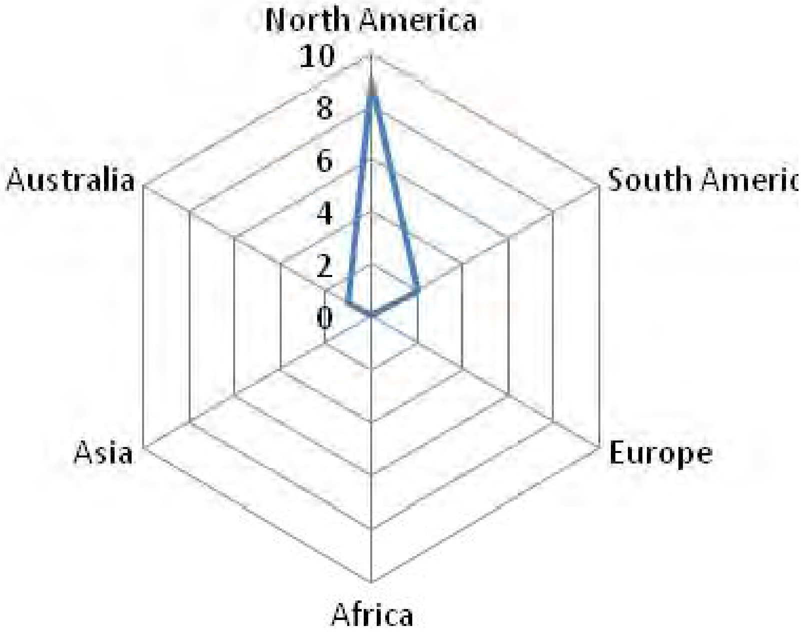
DJFrailey@Raytheon.com

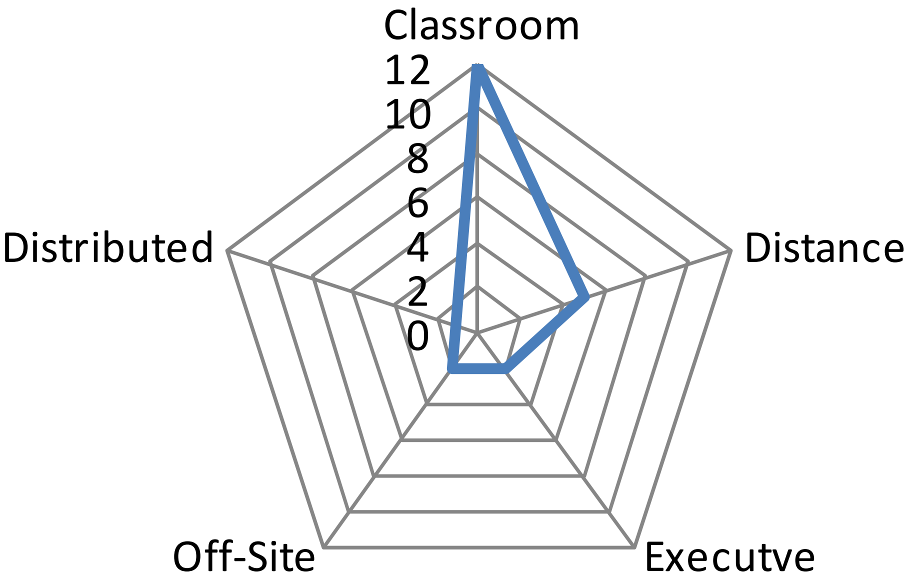
Frailey@lyle.smu.edu

Methodology

- Defined a rubric for comparison of the model with an existing program
- Recruited programs willing to participate in the comparison process
  - Confidentiality was required for many
- Established a method of recording and analyzing the data
- Established norms and procedures for maintaining confidentiality
- Conducted interviews with various university representatives to gather information
- Learned a lot of lessons
  - What data to collect
  - How to collect it
- Improved our methods and procedures

Demographics

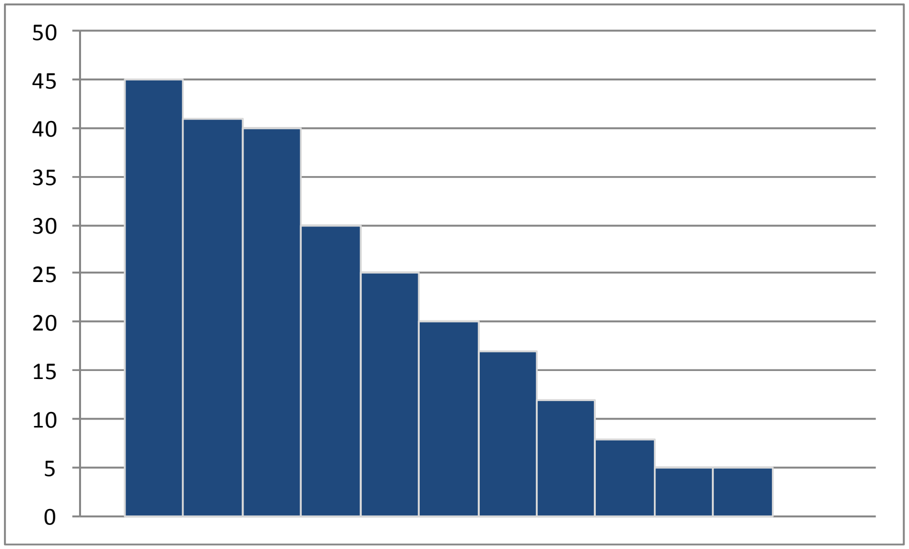




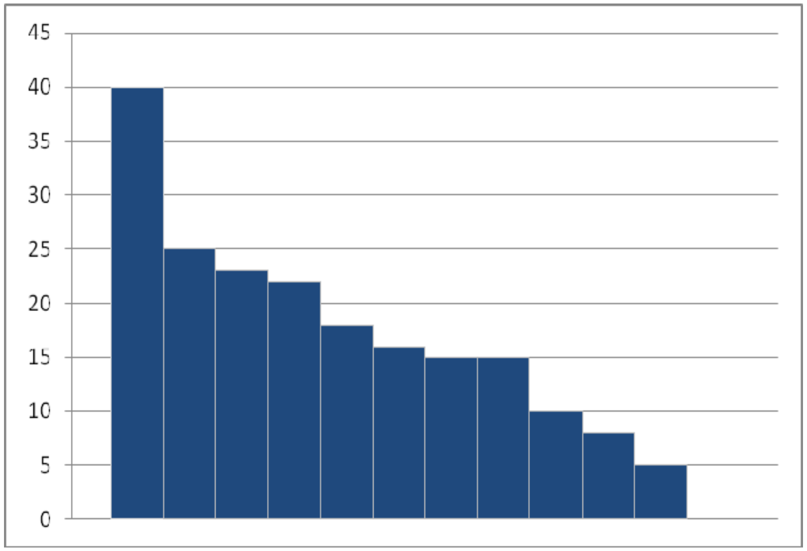
Geographic distribution of participating organizations

Utilization of course delivery methods across participating programs

Program Size

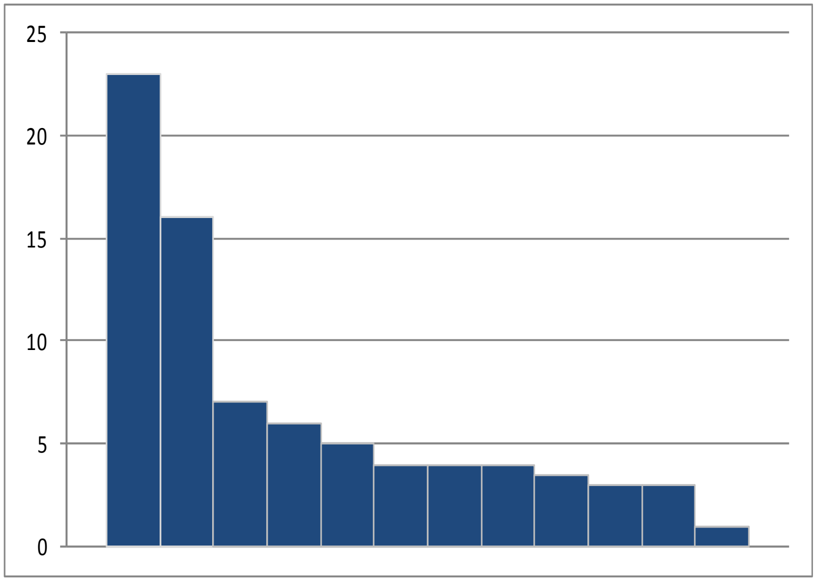


Students graduating per year

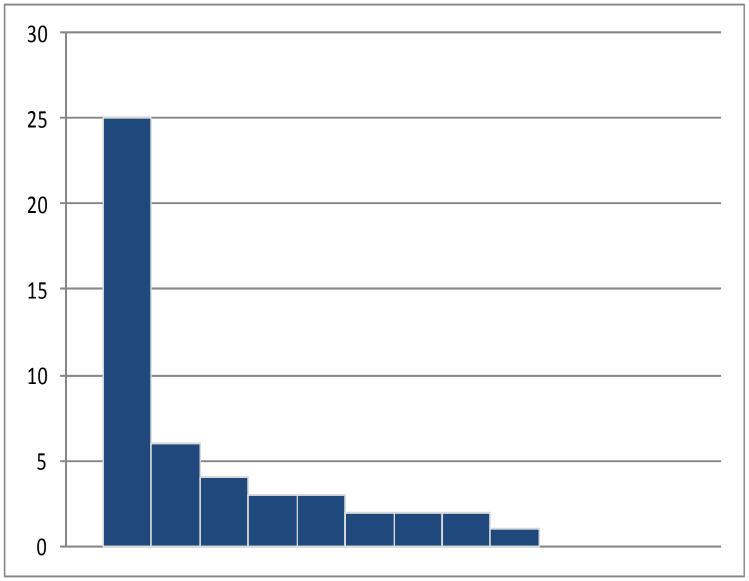


Number of SW Eng Courses

Program Size by # of Faculty

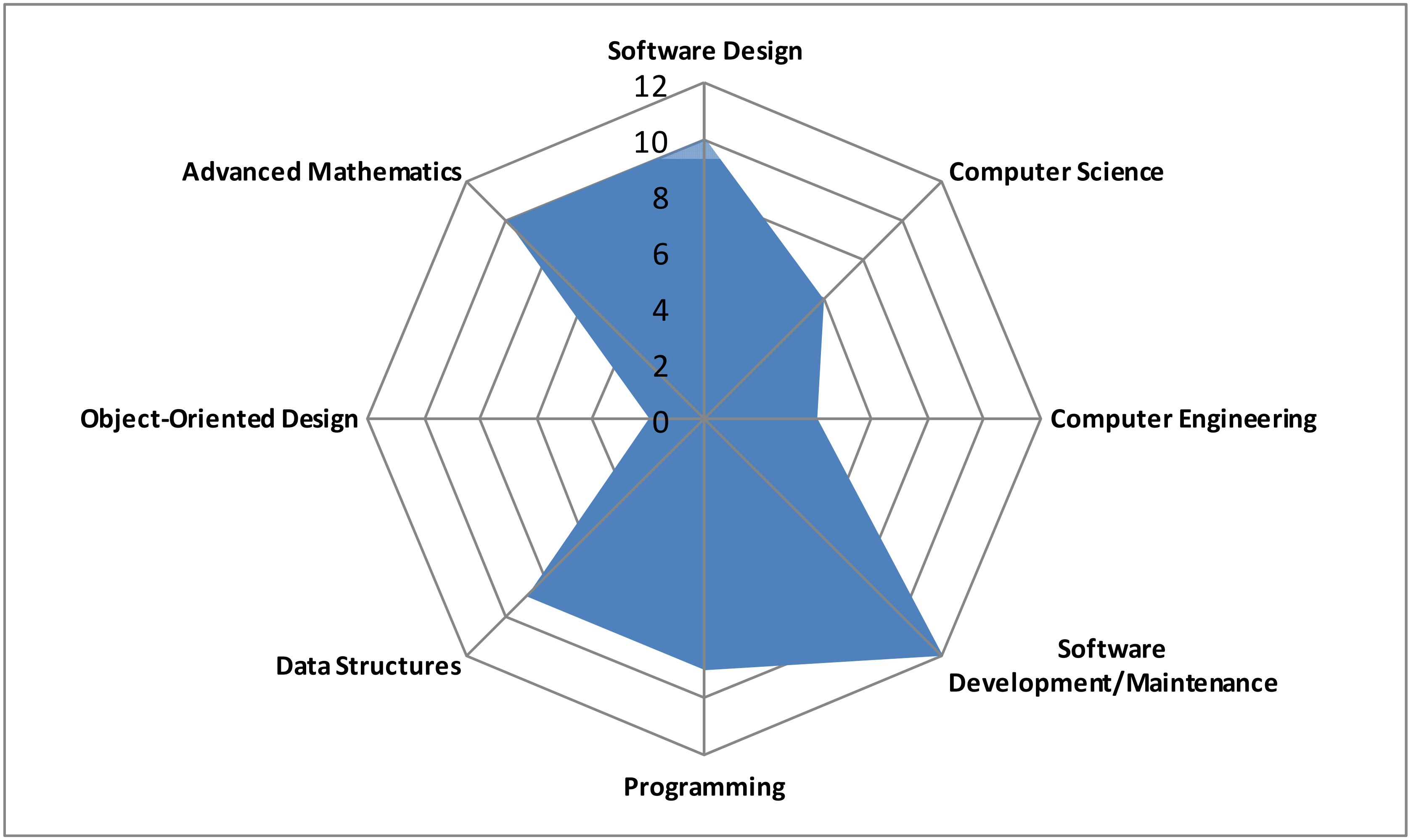


Full Time Faculty

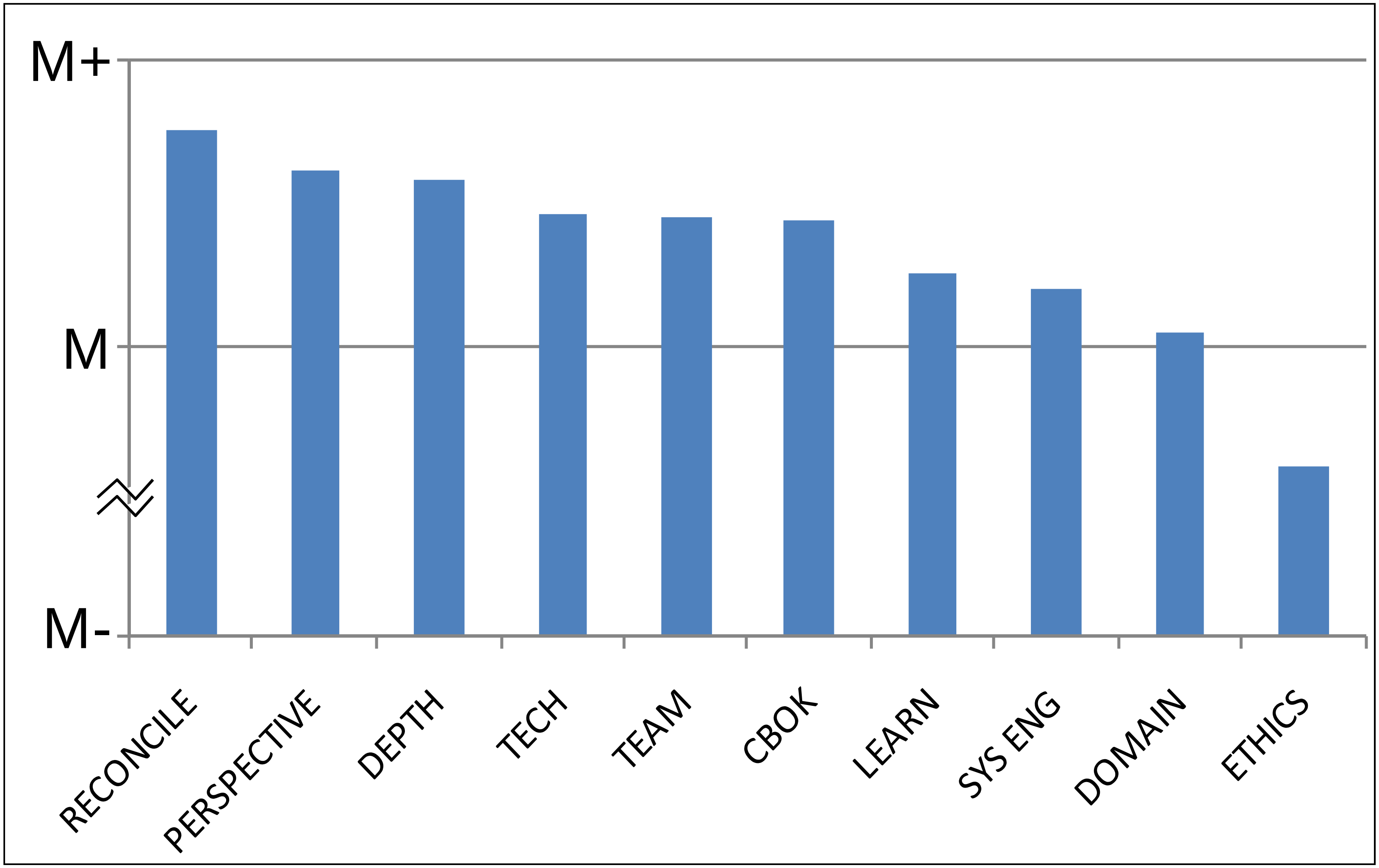


Part Time Faculty

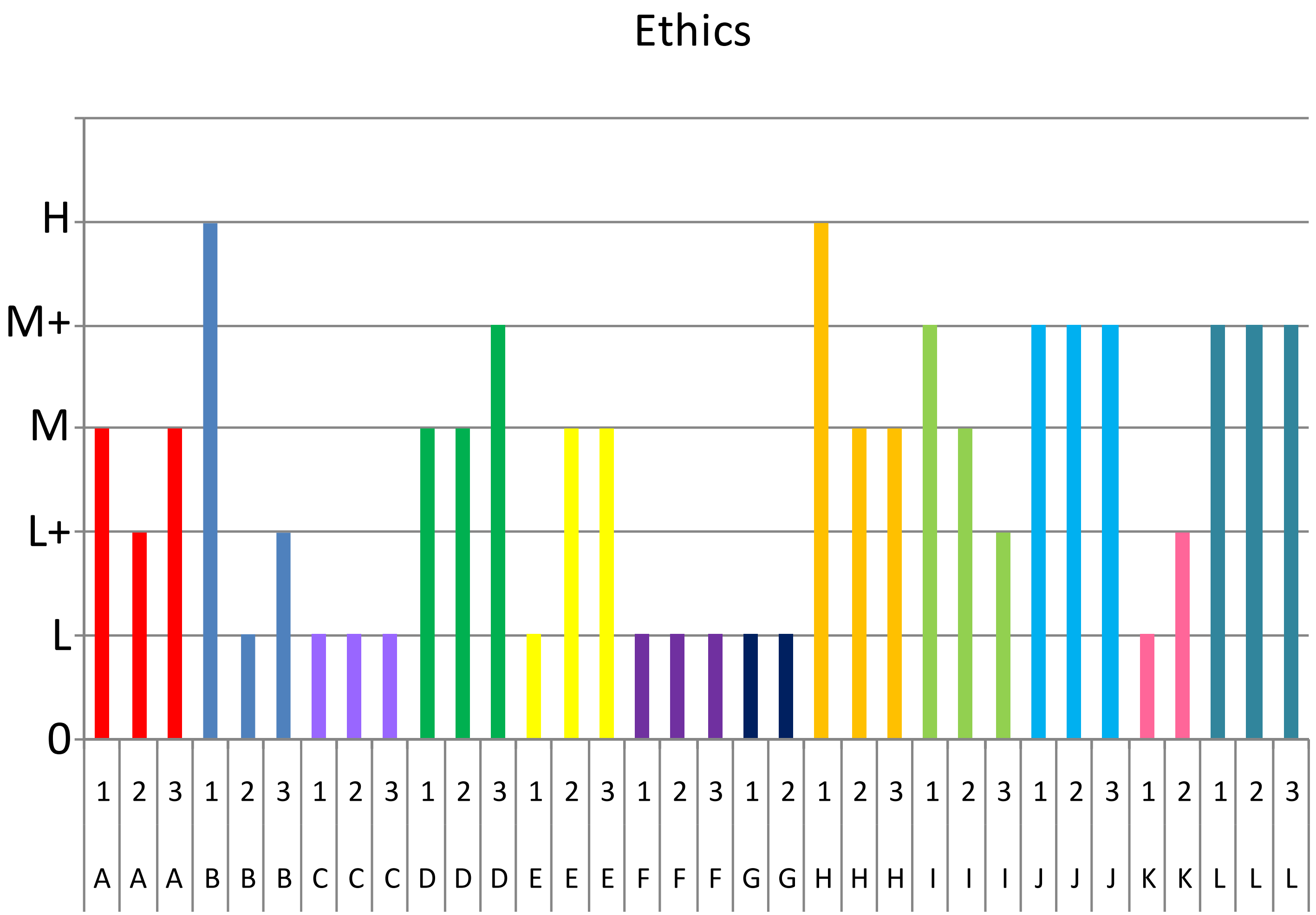
Entrance Requirements



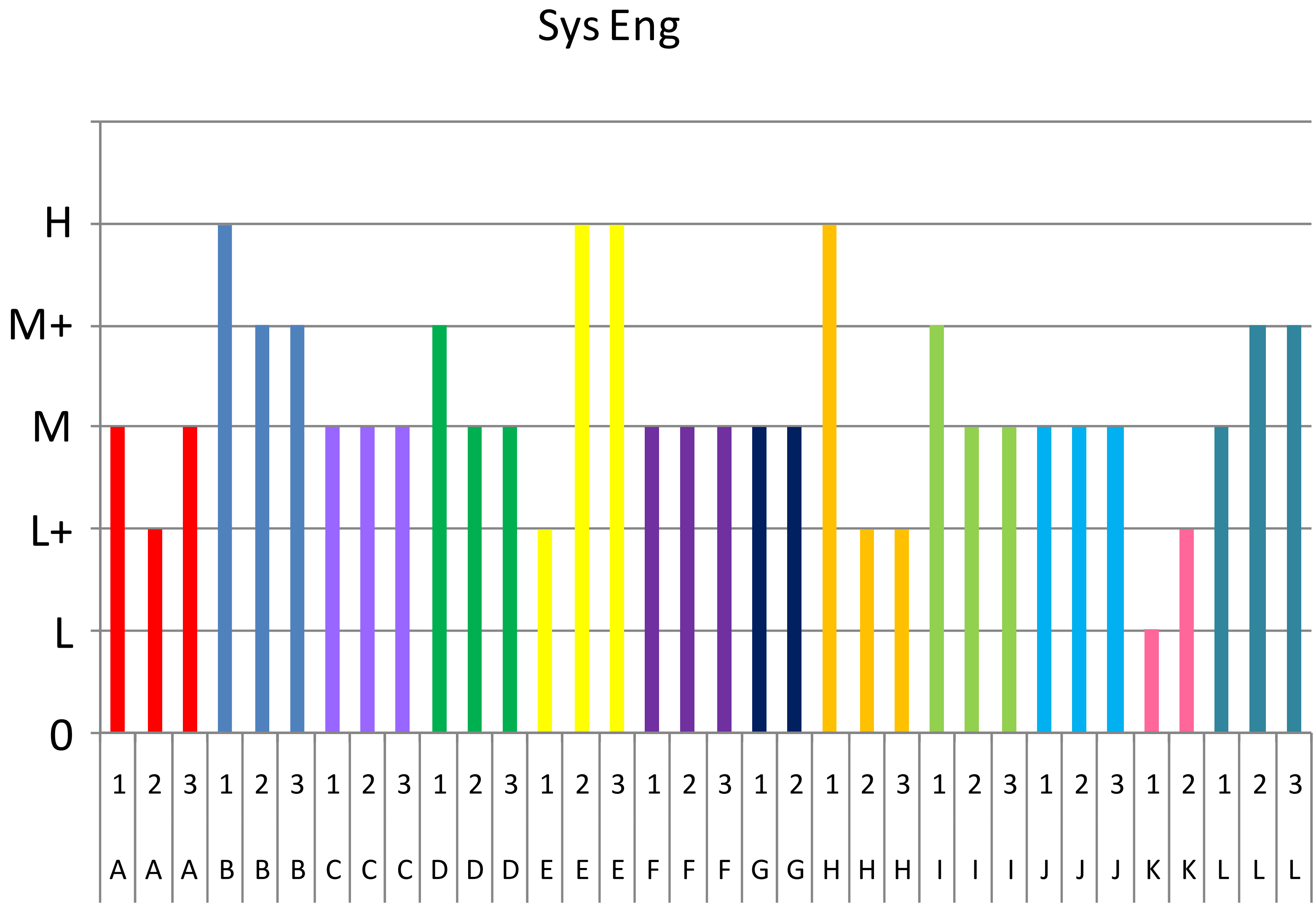
Average Outcome Attainment



Variation in Attainment (Showing Ethics Outcome)



Systems Engineering Outcome Attainment



Next Steps

- Continued Evaluation of Existing Programs vs the Model
  - – We’ve learned a lot about what kind of information we need
  - – International institutions often have trouble relating to the model
- Refinement of the Model and the Data Collection Process
  - – Based on what we are learning
- Expanding the number of Programs to more international locations

Transition to Bre[illegible]ichael – Implementation Issues

Implementation FAQ Categories

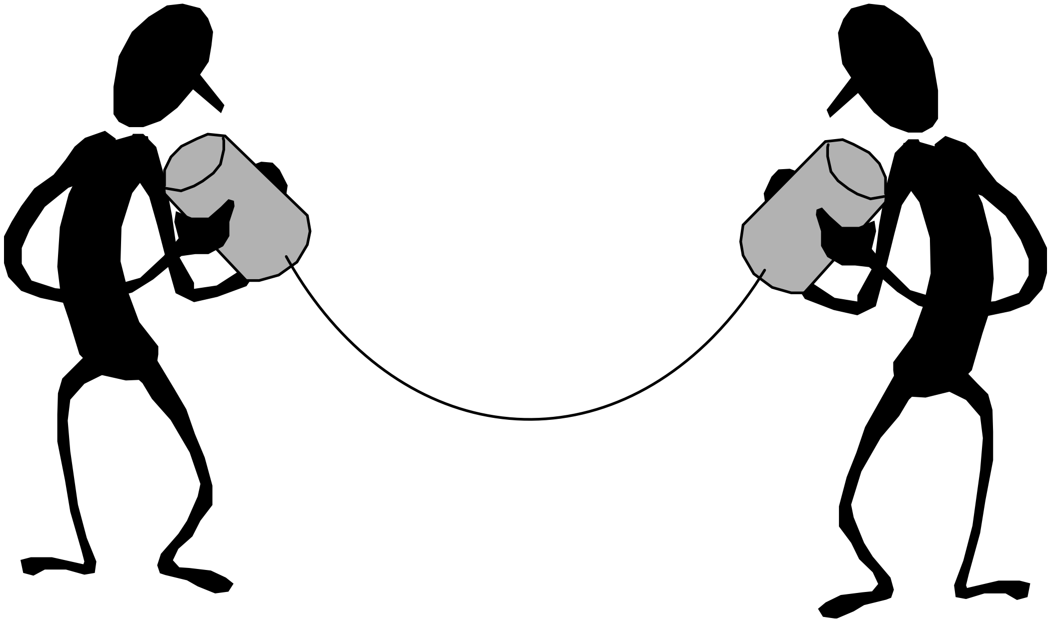
- Planning
- Internal Communication
- Acquiring Resources
- External Communication
- Implementation
- Program Evolution

Planning

- Identify Champions
  - Internal
  - External
- Identify the Customers and the Market
  - Communicate with potential customers
- Identify potential Employers of Graduates
  - Make them champions
- Start-up Funding
- University Political Issues
  - Where the program fits in the university
- Getting Adequate Faculty Resources
  - Adjuncts from Potential Employers
  - Other Departments in the University

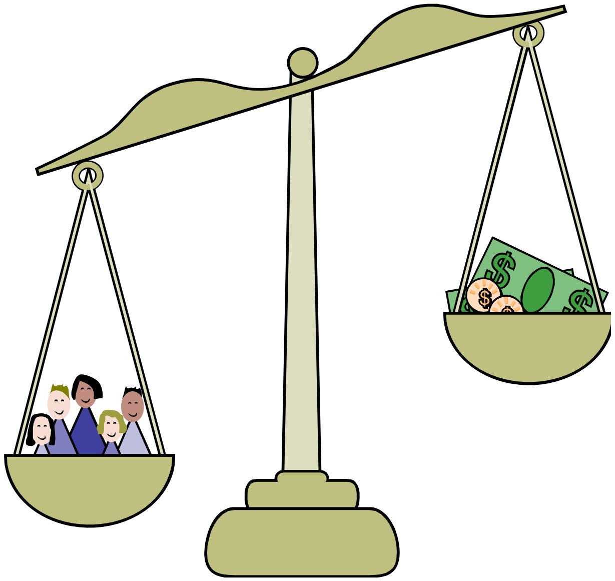
Internal Communication

- Internal Stakeholders
  - Identify them
  - Communicate with them
  - Market to them
  - Get them on your side



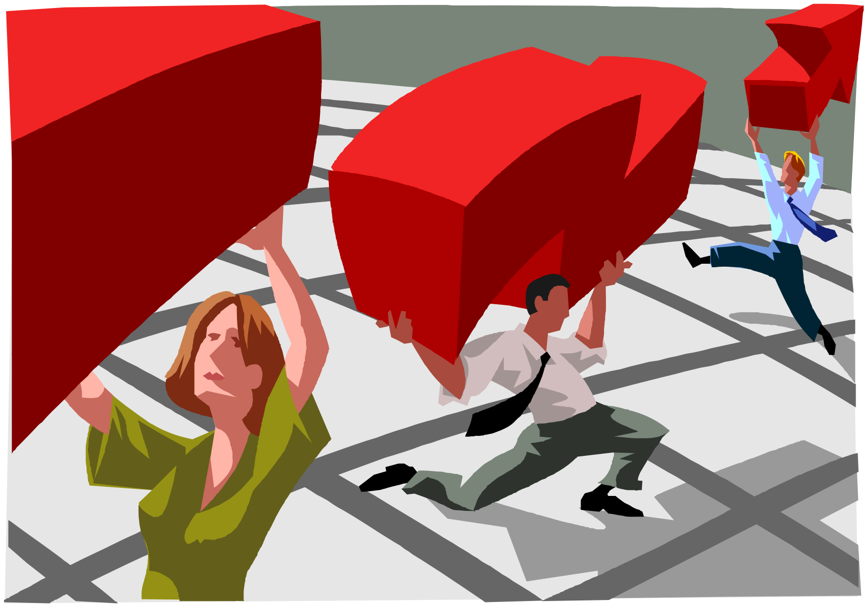
Acquiring Resources

- Faculty Resources
  - – Cast a wide net
  - – Salary differences
- Hardware and Software Resources
- Courseware
- Space



External Communication

- Marketing to Potential Students
- Industrial Champions
- Employers
- Community Visibility



Implementation

- Control of Curriculum
- Integration of External Faculty
- Pedagogical Issues
- Making Up for Student Deficiencies upon Entry
- Handling Students with Advance Preparation
- Opportunities for Students to Gain Work Experience
- Joint Undergraduate/Graduate Programs
- Relying on Other Academic Units
- Handling Cross-Cutting Topics
- Innovative Delivery Options
  - Distance Education
  - Executive Format
  - Etc.

Program Evolution

- Teaming with Other Universities
  - To get a quick start
- Reducing Dependence on Other Universities
  - Once you have built up the resources
- Sustaining Quality
- Adapting to Change
  - Technology
  - Curriculum
- Correcting Program Problems

Questions