The *Integrated Software and Systems Engineering Curriculum Project:* Creating a Reference Curriculum for Graduate Software Engineering Education

Kristen Baldwin and Art Pyster

October 23, 2007

Office of the Under Secretary of Defense
Acquisition, Technology and Logistics
Systems and Software Engineering

Stevens Institute of Technology
School of Engineering
Applied Systems Thinking Institute

Background

- Software drives the performance of virtually all major systems.
- Being able to produce software that can be trusted as reliable, secure, safe, correct, and available while being delivered on-time and within budget is a major challenge for both the government and industry.
- Many steps must be taken to meet that challenge - including ensuring our workforce is well educated in software engineering (SWE) principles and practices.
- Yet today, there is no commonly accepted modern structure or content for graduate software engineering education. Last effort was in early 1990s by the SEI.

iSSEc **- The Way Forward**

The *Integrated Software and Systems Engineering Curriculum Project (iSSEc)* is creating a reference curriculum leading to a Masters degree in software engineering

iSSEc - The Way Forward

- *iSSEc* is sponsored by DOD and led by Stevens, involving 4 sets of stakeholders:
  - The industrial and government workforce who are the customers of SWE graduate education
  - Academics who provide SWE and SE graduate education
  - Professional societies with a vested interest in SWE and SE graduate education
  - Government organizations who fund improvements in SWE graduate education
- *iSSEc* recognizes that the divide between systems and software engineers in industry, government, and academia works against successfully delivering modern systems in which software is almost always central.
- *iSSEc* will integrate SE principles and practices into the SWE curriculum. The bright line that now separates SE and SWE in academia must be eliminated!

The Approach

1. Understand the current state of SWE graduate education (November 2007)
2. Create a *strawman* model curriculum, suitable for broad use, with a small representative team (February 2008)
3. Publicize effort through conferences, papers, website, etc. (continuous)
4. Gradually obtain endorsement from ACM, IEEE, INCOSE, NDIA, and other professional organizations (continuous)
5. Create full model curriculum, suitable for global use, with a large representative team (September 2008 and September 2009)
6. Seek early adopters (continuous)

Status - Understand Current State

1. Understand the current state of SWE graduate education (November 2007)
2. Create a *strawman* model curriculum, suitable for broad use, with a small representative team (February 2008)
3. Publicize effort through conferences, papers, website, etc. (continuous)
4. Gradually obtain endorsement from ACM, IEEE, INCOSE, NDIA, and other professional organizations (continuous)
5. Create full model curriculum, suitable for global use, with a large representative team (September 2008 and September 2009)
6. Seek early adopters (continuous)

Understanding the Current State

- Select diverse set of universities with Masters programs in SWE - vary in size, geography, maturity, resources, target market, ...
- Use Software Engineering Body of Knowledge (SWEBOK) as primary framework for SWE competencies
- Collect data from school websites
  - Degree, faculty size, student population, target market, ...
  - Degree structure, individual course descriptions
  - Map between courses and SWEBOK
- Validate data with professor
- Analyze for commonalities and uniqueness

Schools Completed or In Process

1. Air Force Institute of Technology
2. Brandeis University
3. California State University - Fullerton
4. California State University-Sacramento
5. Carnegie Mellon University
6. Carnegie Mellon University West
7. Carrol College
8. DePaul University
9. Dublin City University (Ireland)
10. Embry-Riddle Aeronautical University
11. Florida A&M
12. George Mason University
13. James Madison University
14. Kingston University (UK)
15. Mercer University
16. Monmouth University
17. Naval Postgraduate School
18. Rochester Institute of Technology
19. Seattle University
20. Southern Methodist University
21. Stevens Institute of Technology
22. Texas Tech
23. University of Alabama-Huntsville
24. University of Colorado - Colorado Springs
25. University of Michigan - Dearborn
26. University of Quebec (Canada)
27. University of Scranton
28. University of Southern California
29. University of Sunderland (UK)
30. University of York (UK)

Some changes still likely

SWEBOK’s 10 Knowledge Areas

REQ	Software Requirements
DES	Software Design
CST	Software Construction
TST	Software Testing
MNT	Software Maintenance
CNF	Software Configuration Management
MGT	Software Engineering Management
PRC	Software Engineering Process
TLS	Software Engineering Tools and Methods
QLY	Software Quality

Early Observations from 11 Schools

1. SWE is largely viewed as a specialization of Computer Science - much as systems engineering was often viewed as specialization of industrial engineering or operations research years ago
2. Faculty size is small - few dedicated SWE professors, making programs relatively fragile
3. Student enrollments are generally small compared to CS and to other engineering disciplines
4. Many programs specialize to specific markets such as defense systems or safety critical systems
5. The target student population varies widely - anyone with Bachelors and B average to someone with CS degree and 2+ years of experience

More Early Observations

6. Program outcomes vary widely - software developer to researcher to software manager
7. Wide variation in depth and breadth of SWEBOK coverage in required and semi-required courses
8. SWEBOK alone does not represent the breadth of many program's required courses
9. Some significant topics are rarely mentioned - agility, Software Engineering Economics, Systems Engineering
10. Some topics are ubiquitous - formal methods and architecture
11. "Object-Oriented" is the standard development paradigm - creating a "clash" with many systems engineering programs that emphasize structure methods

Sample Program Specialty

Air Force Institute of Technology	Defense Systems
Embry-Riddle Aeronautical University	Embedded Real-time Software
Naval Postgraduate School	Acquisition of Defense Systems
Seattle University	Project Experience
Stevens Institute of Technology	Quantitative Software Engineering
University of Southern California	Quantitative; Software Economics
University of York (UK)	Safety Critical Systems

Sample Program Focus

Air Force Institute of Technology	Develop professionals to develop and manage increasingly complex software
Embry-Riddle Aeronautical University	How to engineer high-performance sofwtare embedded in aircraft, space and medical systems
George Mason University	Developing and modifying large, complex software systems. Emphasis both technical and management aspects
Monmouth University	Effective member of software development team
Naval Postgraduate School	Enable acquisition professionals to procure highly dependable, trustworthy software-intensive systems
Seattle University	Understand and apply advanced software engineering principles vital to industry
Stevens Institute of Technology	Realizing software products on time, within budget and with known quality
University of Alabama – Huntsville	Provide fundamentals of software development for members of software development teams
University of Southern California	Prepare students for an industrial leadership career in software engineering and serve as introduction to researchers
University of York (UK)	Software systems with a high requirement for dependability.

Sample Target Student

Air Force Institute of Technology	PMs and software developers from DoD & other agencies
California State University – Sacramento	UG degree with CS courses
Embry-Riddle Aeronautical University	Strong academic record
George Mason University	UG degree
Monmouth University	UG degree in CS, SWE or Engineering related
Naval Postgraduate School	Acquisition professionals with 2+ years in software development
Seattle University	UG degree in CS or equivalent 2+ years in software development
Stevens Institute of Technology	Experienced computer professionals seeking leadership positions
University of Alabama – Huntsville	UG courses in CS, math & statistics
University of Southern California	UG degree in CS, math or engineering with courses in computing and math
University of York (UK)	UG degree in CS or related field with math

Introduction to Software Engineering

Schools

AFIT	Air Force Institute of Technology	CSE 481
CSUS	California State University - Sacramento	
ERAU	Embry-Riddle Aeronautical University	
GMU	George Mason University	-
MMU	Monmouth University	SE 504
NPS	Naval Postgraduate School	SW 3460
SEA	Seattle University	
SIT	Stevens Institute of Technology	CS 540
UAL	University of Alabama - Huntsville	CS 650
USC	University of Southern California	CS 577a, CS 577b
YOR	University of York (UK)	-

SWEBOK

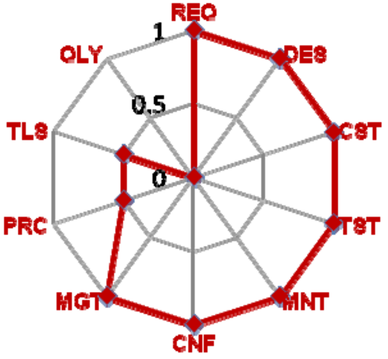
REQ	Software Requirements
DES	Software Design
CST	Software Construction
TST	Software Testing
MNT	Software Maintenance
CNF	Software Configuration Management
MGT	Software Engineering Management
PRC	Software Engineering Process
TLS	Software Engineering Tools and Methods
QLY	Software Quality

Scale

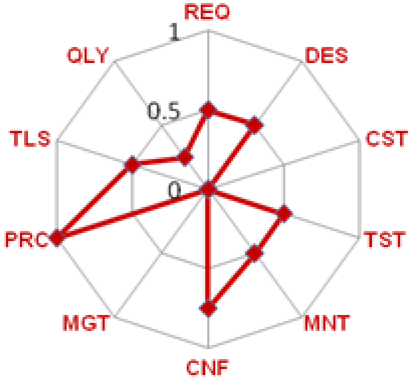
1.00	**> 90% of subtopics**
0.75	**~75% of subtopics**
0.50	**~50% of subtopics**
0.25	**~25% of subtopics**
0.00	**No Coverage**

Introduction to Software Engineering

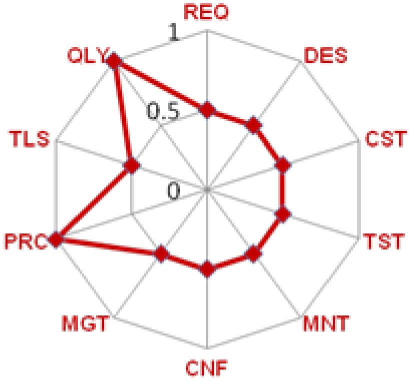
Air Force Institute of Technology



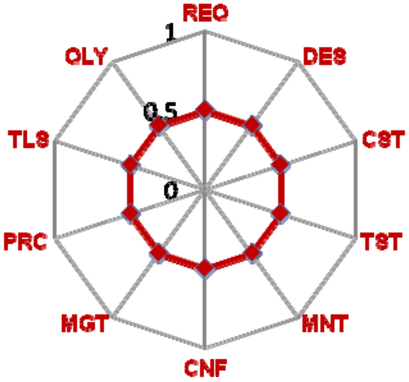
California State University - Sacramento



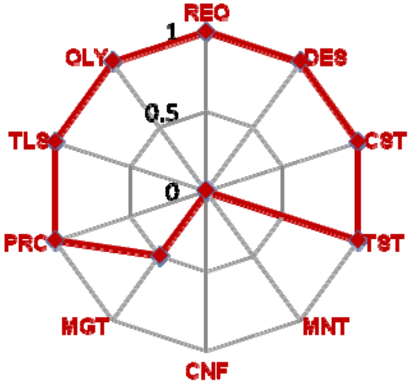
Embry Riddle Aeronautical University



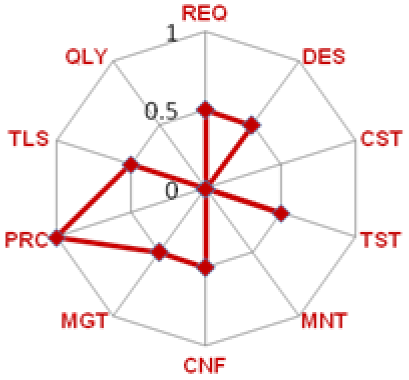
Monmouth University



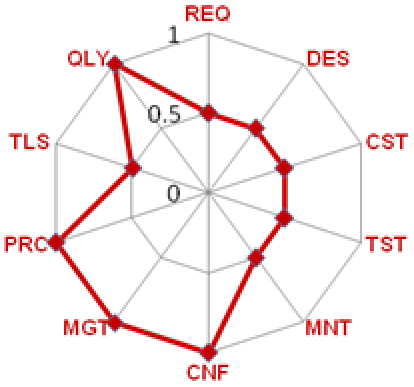
Naval Postgraduate School



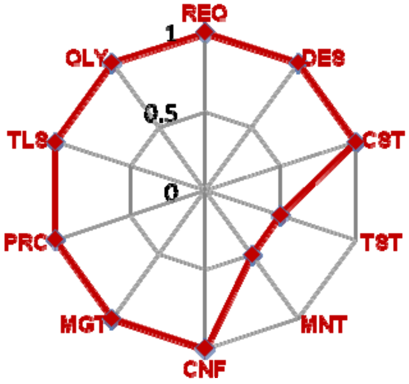
Stevens Institute of Technology



University of Alabama - Huntsville



University of Southern California



1.00 > 90% of subtopics 0.75 ~75% of subtopics 0.50 ~50% of subtopics
0.25 ~25% of subtopics 0.00 No Coverage

Required and Semi-Required Courses

Schools

AFIT	Air Force Institute of Technology
CSUS	California State University - Sacramento
ERAU	Embry-Riddle Aeronautical University
GMU	George Mason University
MMU	Monmouth University
NPS	Naval Postgraduate School
SEA	Seattle University
SIT	Stevens Institute of Technology
UAL	University of Alabama - Huntsville
USC	University of Southern California
YOR	University of York (UK)

Scale

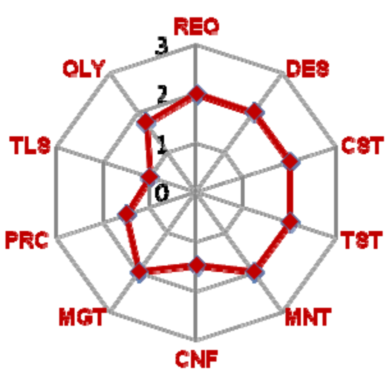
3.00	>1Req. or Semi Req. Course
2.00	1 Req. or Semi Req. Course
1.00	Introductory course
0.00	No Course

SWEBOK

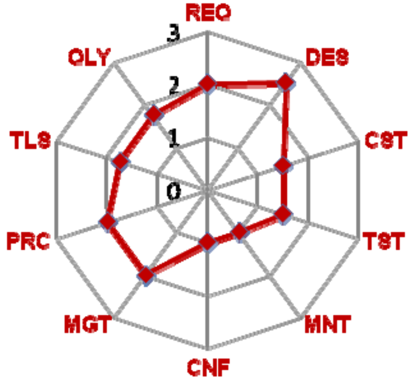
REQ	Software Requirements
DES	Software Design
CST	Software Construction
TST	Software Testing
MNT	Software Maintenance
CNF	Software Configuration Management
MGT	Software Engineering Management
PRC	Software Engineering Process
TLS	Software Engineering Tools and Methods
QLY	Software Quality

Required and Semi-Required Courses

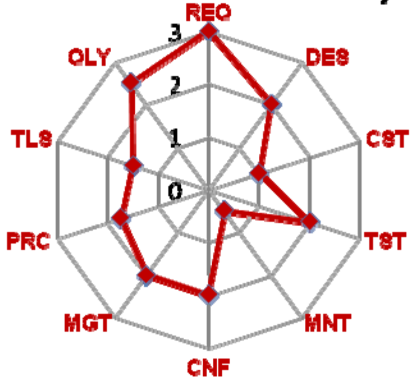
Air Force Institute of Technology



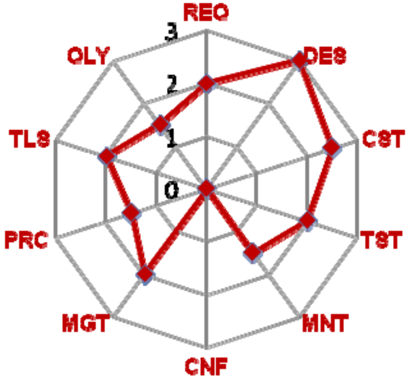
California State University - Sacramento



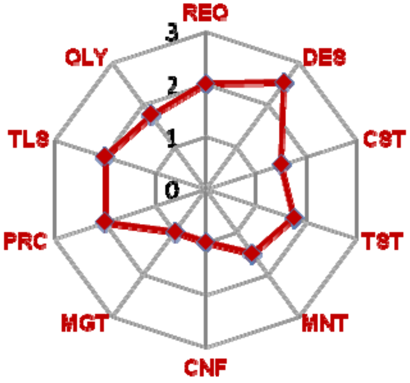
Embry Riddle Aeronautical University



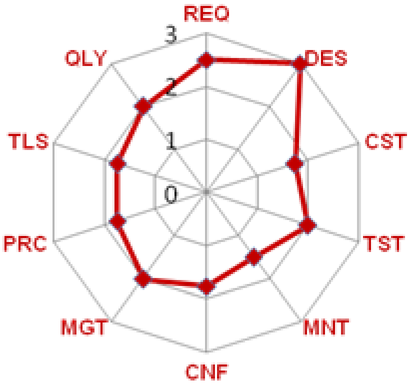
George Mason University



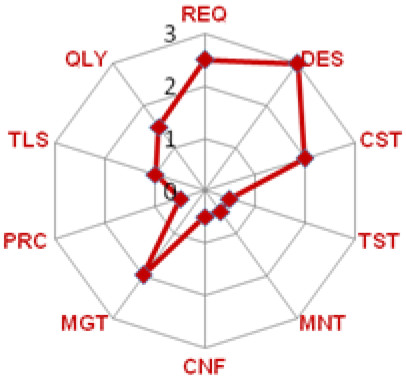
Monmouth University



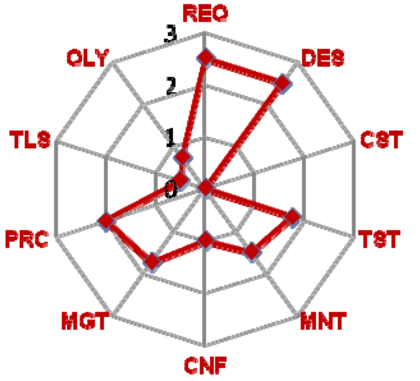
Naval Postgraduate School



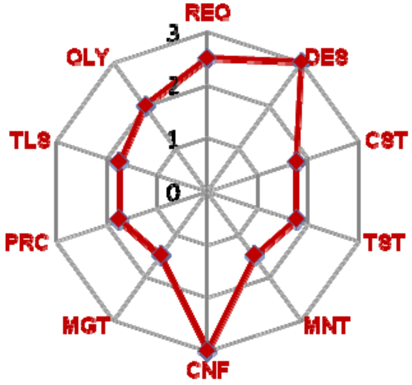
Seattle University



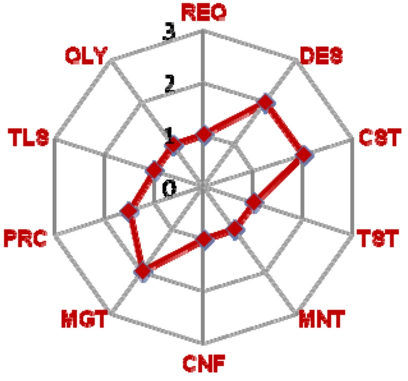
Stevens Institute of Technology



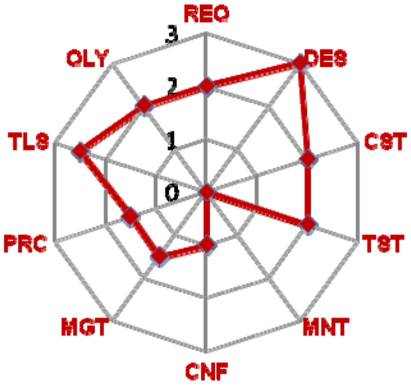
University of Alabama - Huntsville



University of Southern California



University of York



3.00	>1Req. or Semi Req. Course
2.00	1 Req. or Semi Req. Course
1.00	Introductory course
0.00	No Course

Non-SWEBOK

Non-SWEBOK

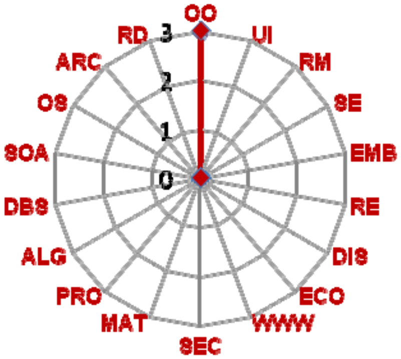
1	**OO**	Object Oriented Systems
2	**UI**	User Interface / human computer interaction
3	**RM**	Research Methodology
4	**SE**	Systems Engineering
5	**EMB**	Embedded & realtime software systems
6	**RE**	Software Reliability
7	**DIS**	Distributed Software Engineering
8	**ECO**	Software Engineering Economics
9	**WWW**	SwE for worldwide web
10	**SEC**	Software Safety & Security
11	**MAT**	Math foundations of SwE
12	**PRO**	Programming
13	**ALG**	Algorithms
14	**DBS**	Database Systems
15	**SOA**	Service Oriented Architecture
16	**OS**	Operating Systems
17	**ARC**	Computer Architecture
18	**RD**	Software R&D

Scale

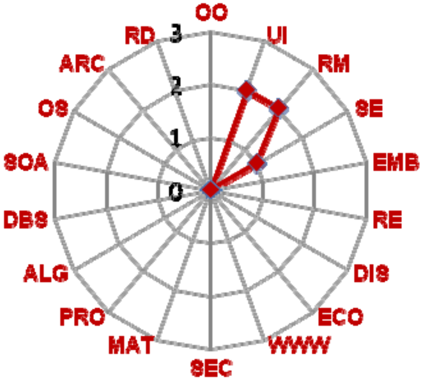
3.00	**More than 1 full course**
2.00	**Full Course**
1.00	**Partial Course**
0.00	**No Course**

(Required and Semi-Required Courses)

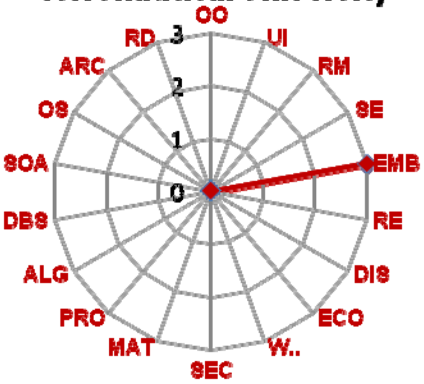
Air Force Institute of Technology



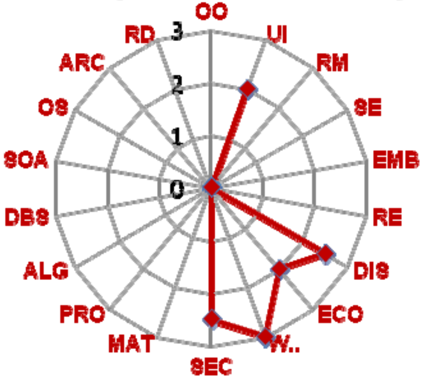
California State University - Sacramento



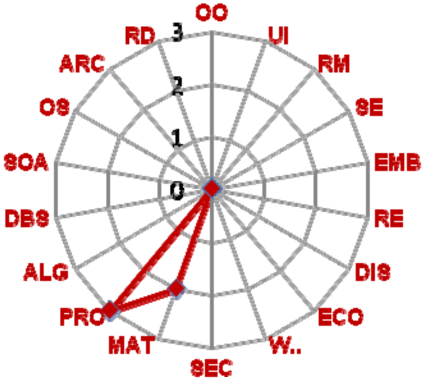
Embry Riddle Aeronautical University



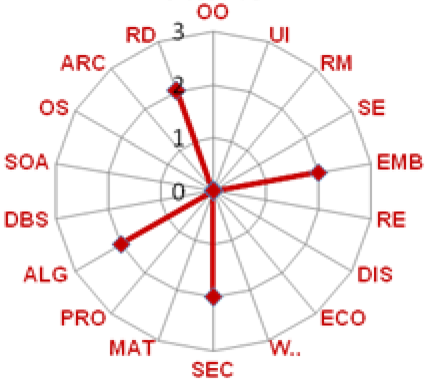
George Mason University



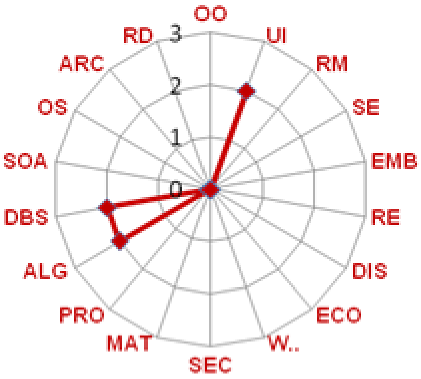
Monmouth University



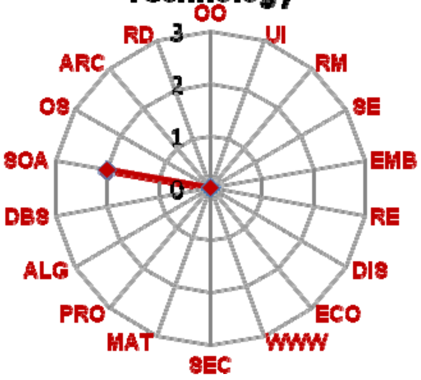
Naval Postgraduate School



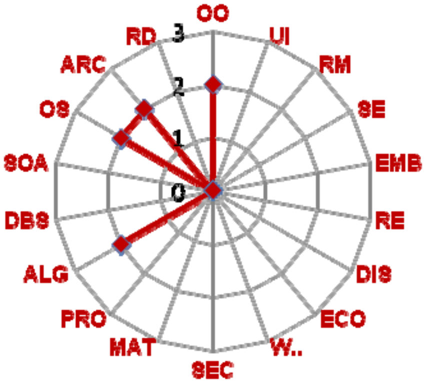
Seattle University



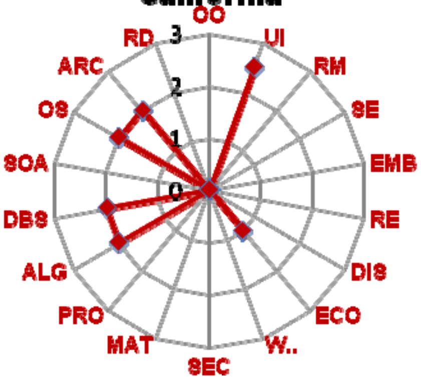
Stevens Institute of Technology



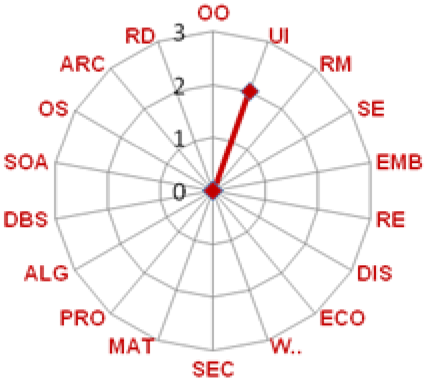
University of Alabama - Huntsville



University of Southern California



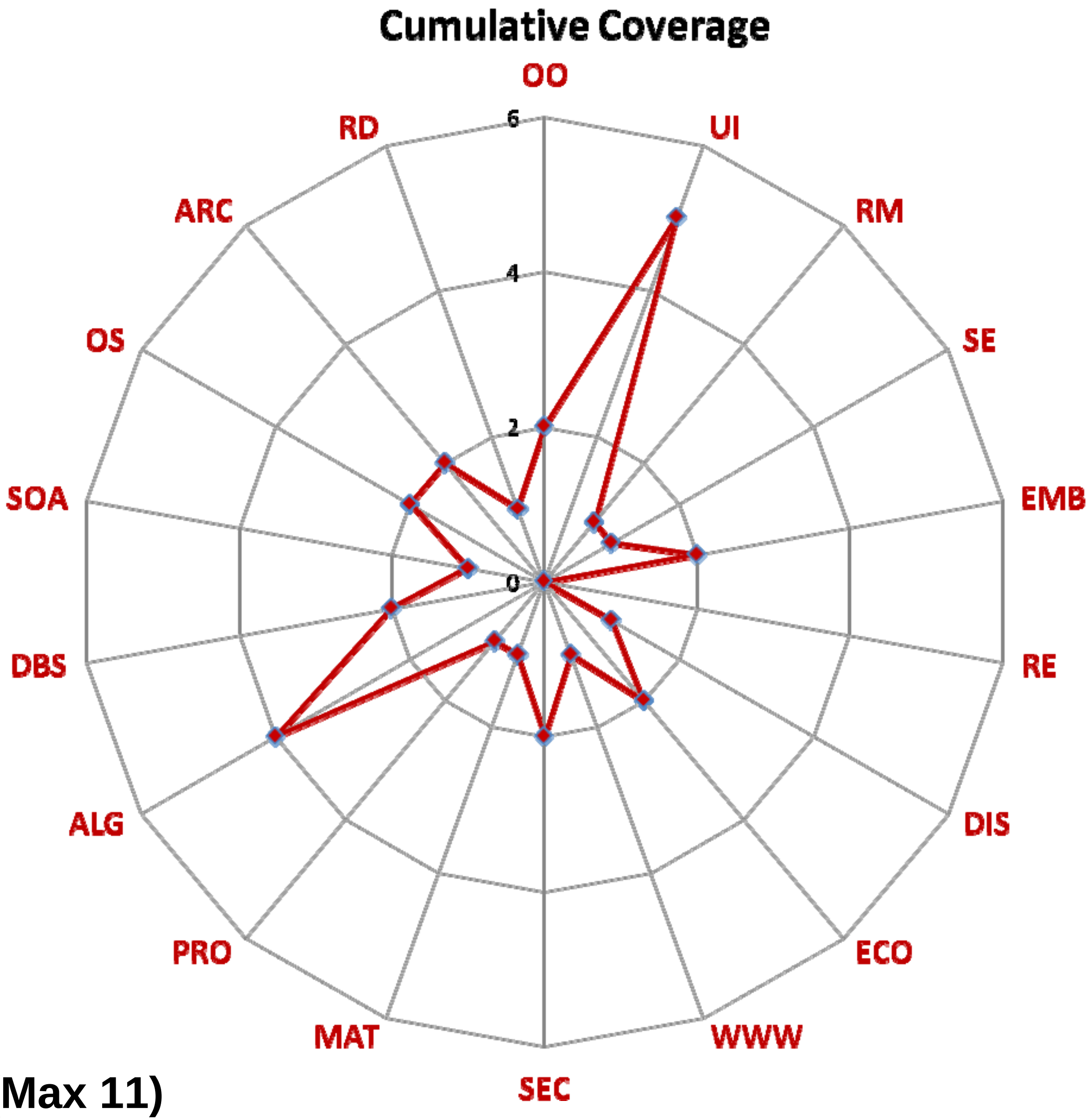
University of York



3.00	More than 1 full course
2.00	Full Course
1.00	Partial Course
0.00	No Course

Non-SWEBOK

Non-SWEBOK



MODE
No. of Schools (Max 11)

Early Start Team Members

1. Bruce Amato, *Department of Defense*
2. Mark Ardis*, RIT*
3. Larry Bernstein, *Stevens*
4. Barry Boehm, *USC*
5. John Brackett, Boston University
6. Murray Cantor, *IBM*
7. Robert Edson, *ANSER*
8. Gary Hafen, *NDIA and Lockheed Martin*
9. Tom Hilburn, *Embry-Riddle Aeronautical University*
10. Jim McDonald, *Monmouth University*
11. Ernest McDuffie*, National Coordinating Office*
12. Bret Michael, *NPS*
13. Bill Milam, *Ford*
14. Ken Nidiffer, *SEI*
15. Art Pyster, *Stevens*
16. Paul Robitaille, *INCOSE and Lockheed Martin*
17. Doug Schmidt, *Vanderbilt*
18. Mary Shaw, *Carnegie Mellon University*
19. Richard Thayer, *California State University at Sacramento*
20. Richard Turner, *Stevens*
21. Osmo Vikman, *Nokia*
22. David Weiss, *Avaya*

Graduate Students:

- Deva Henry
- Kahina Lasfer
- Sarah Sheard

Observer:

- Joe Urban, NSF
- Lillian Cassel, ACM

Several more offers and lots of interest…

Status - Create Strawman Curriculum

1. Understand the current state of SWE graduate education (November 2007)
2. Create a *strawman* model curriculum, suitable for broad use, with a small representative team (February 2008)
3. Publicize effort through conferences, papers, website, etc. (continuous)
4. Gradually obtain endorsement from ACM, IEEE, INCOSE, NDIA, and other professional organizations (continuous)
5. Create full model curriculum, suitable for global use, with a large representative team (September 2008 and September 2009)
6. Seek early adopters (continuous)

Creating the Strawman Curriculum

1. Held workshop on August 15-16 at Applied Systems Thinking Institute
2. Reviewed foundational documents: SEI graduate curriculum reports from 1991, SWEBOK, SE2004, INCOSE SE Model Graduate Curriculum
3. Agreed to create strawman curriculum and agreed on outline of document
4. Divided into 4 primary teams with leads from 4 different universities
    - Guidance and Outcomes - Art Pyster, Stevens Institute
    - Curriculum Architecture - Jim MacDonald, Monmouth
    - Body of Knowledge - Tom Hilburn, Embry-Riddle
    - Course Packaging - Brett Michael, Naval Postgraduate School
5. Agreed to work in parallel where possible to speed delivery

Creating the Strawman Curriculum

6. Build Guidance and Outcomes as deltas from SE2004 Principles (Draft 1 done)
7. Build Architecture starting with 1991 SEI curriculum architecture (Draft 1 under review)
8. Build Body of Knowledge as deltas from SWEBOK using INCOSE Handbook, PMI BOK, and current state of SWE graduate programs as primary sources for additions (Draft 1 begun)
9. Build Course Packaging after first three teams have solid drafts
10. Hold second workshop in December to review progress
11. Refine drafts and publish at end of February

Sample Draft Guidance

Software Engineering draws its foundations from a wide variety of disciplines.

- Graduate study of software engineering relies on many areas in computer science for its theoretical and conceptual foundations, but it also draws from other fields, including statistics, logic, calculus, discrete mathematics, formal languages, and other mathematical specialties, from systems and domain engineering, from project and portfolio management, and from one or more application domains.

MSwE2008 must identify prerequisite requirements for students to enter an MSE program.

- Undergraduate computing programs and industry experience in software engineering vary greatly. To help institutions build programs that address the needs of the broad software engineering community, MSwE2008 recommends minimum prerequisite knowledge necessary to successfully engage in a program based on the MSwE2008 curriculum. Generally, that knowledge comes from a technical, scientific, or engineering undergraduate degree including coursework in computer science. However, relevant work experience can substitute for formal education. Schools that wish to admit students lacking that minimum prerequisite knowledge should provide preparatory courses that those students should take before entering the Masters program.

Sample Draft Outcome

Show mastery of the software engineering knowledge and skills, and professional issues necessary to practice as a software engineer in a variety of application domains with demonstrated performance in at least one application domain.

Students, through regular reinforcement and practice, need to gain confidence in their abilities as they progress through a software engineering program of study. At graduation, a student should understand what distinguishes practice in different application domains such as finance, medical, transportation, and telecommunications, should understand how to learn a new domain as needed, and should demonstrate skill as a software engineer in at least one application domain. Such demonstration will include (as defined in Bloom's Taxonomy)

- At least comprehension level competency across all MSwE2008 BOK knowledge areas, not including the KA on "Knowledge Areas of the Related Disciplines".
- Application level competency, or above, in 75% of the MSwE2008 BOK knowledge areas.

Hence, a graduate should be able to analyze, design, verify, validate, implement, apply, and maintain a modest-sized software system and understand the challenges of scaling to larger software systems. In addition, graduates need to have gained an understanding and appreciation of professional issues related to ethics and professional conduct, economics, and the societal needs.

Sample Draft Outcome

Work effectively as part of a team, including teams that may be international and geographically distributed, to develop quality software artifacts, and to lead in one area of project development, such as project management, requirements analysis, architecture, construction, or quality assurance.

Students need to complete tasks that involve work as an individual, but also many other tasks that entail working with a group of individuals. For group work, students ought to be informed of the nature of groups and of group activities/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, communication, and individual as well as team performance evaluations. Students should have an appreciation of team dynamics and leadership techniques and be able to lead at least one of the areas. Increasingly, teams are assembled from many geographical sites, often across national boundaries. This presents additional challenges of time, language, and culture that students must know how to address.

Status - Obtain Endorsement

1. Understand the current state of SWE graduate education (November 2007)
2. Create a *strawman* model curriculum, suitable for broad use, with a small representative team (February 2008)
3. Publicize effort through conferences, papers, website, etc. (continuous)
4. Gradually obtain endorsement from ACM, IEEE, INCOSE, NDIA, and other professional organizations (continuous)
5. Create full model curriculum, suitable for global use, with a large representative team (September 2008 and September 2009)
6. Seek early adopters (continuous)

Endorsements

- NDIA SE Division endorsed *iSSEc* in June 2007
- INCOSE Board of Directors endorsed *iSSEc* in October 2007
- ACM Education Board is following *iSSEc* progress and is considering endorsement
- IEEE Computer Society is following *iSSEc* progress and is considering endorsement
- Endorsement from other organizations is possible

Finally…

- The team working on the Strawman Curriculum has been doing a great job and are keeping to the planned schedule
- A workshop among the broad community to review the Strawman Curriculum and to plan the creation of the full curriculum will be held in March or April 2008 - hope to publish another iteration in September 2008 and another in September 2009 that reflects broad community involvement
- Expect a number of early adopters, including schools represented on the Early Start Team that is building the Strawman Curriculum
- Ultimately, *iSSEc* may create a model curriculum for an interdisciplinary degree that fully integrates software and systems engineering graduate education