Educating the Next Generation of Software Engineers

Art Pyster
Distinguished Research Professor
Stevens Institute of Technology and
Deputy Executive Director, SER-UARC

October 22, 2008

art.pyster@stevens.edu

Agenda

- How the world has changed
- The current state of software engineering education
- A new reference curriculum

A History Lesson - 1996

Resolved: Software Should Lead in Systems Engineering

Jim Armstrong vs. Art Pyster

The systems engineering community has long debated the extent to which software disciplines, processes, and practitioners should influence systems engineering. In August 1996, the authors held a lively debate at a meeting of the Washington Metropolitan Area Chapter of INCOSE on the proper role of software engineering within systems engineering. The particular issue debated was the proposal that software ideas, process, and people should be in the lead when building complex systems. Pyster favored that view while Armstrong opposed it.

School of Systems & Enterprises

Software will be the center of systems design.

Eberhardt Rechtin, 1993

Twenty years from now, software people will be sitting at the table and the other disciplines will be sitting around the sides of the room.

Eberhardt Rechtin, 1993

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 SEI in the early 1990s.
- There are, in effect, no current community-endorsed recommendations on what to teach software engineers – nothing that recognizes how the world has changed.
- Response: create a project to create a new reference curriculum in software engineering

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
- Three products planned:
    1. A modern reference curriculum for a master’s degree in software engineering that integrates an appropriate amount of systems engineering
    2. A modern reference curriculum for a master’s degree in systems engineering that integrates an appropriate amount of software engineering
    3. A truly interdisciplinary degree that is neither systems nor software engineering – it is both

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. Obtain endorsement from INCOSE, NDIA, ACM, IEEE, and other professional organizations (continuous)
5. Create GSwERC 0.50 suitable for broad community review and early adoption (October 2008)
6. Create GSwERC 1.0 suitable for broad adoption (2009)

The evolving author team

- Rick Adcock, *Cranfield University and INCOSE*
- Edward Alef, *General Motors*
- Bruce Amato, *Department of Defense*
- Mark Ardis, *Rochester Institute of Technology*
- Larry Bernstein, *Stevens Institute of Technology*
- Barry Boehm, *University of Southern California*
- Pierre Bourque, *Quebec University and SWEBOK volunteer*
- John Bracket, *Boston University*
- Murray Cantor, *IBM*
- Lillian Cassel, *Villanova and ACM volunteer*
- Robert Edson, *ANSER*
- Richard Fairley, *Colorado Technical University*
- Dennis Frailey, *Raytheon & Southern Methodist University*
- Gary Hafen, *Lockheed Martin and NDIA*
- 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*
- 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*
- William Milam, *Ford*
- Ken Nidiffer, *Software Engineering Institute*
- Art Pyster, *Stevens Institute of Technology*
- Doug Schmidt, *Vanderbilt University*
- Mary Shaw, *Carnegie Mellon University*
- Robert Suritis, *IBM*
- Richard Thayer, *California State University at Sacramento*
- Barrie Thompson, *Sunderland University, UK*
- Richard Turner, *Stevens Institute of Technology*
- Joseph Urban, *Texas Technical University*
- Ricardo Valerdi, *MIT & INCOSE*
- David Weiss, *Avaya*
- Mary Jane Willshire, *Colorado Technical University*

Methodology to understand current state of SwE education

- Diverse set of universities with Masters programs in SWE
  - Vary in size, geography, maturity, resources, target market, …
  - Focused on programs with degree in SWE or Computer Science with a SWE specialization - not degrees in information technology and related areas
- Used Software Engineering Body of Knowledge (SWEBOK) as the primary framework for SWE competencies
- Collected data from school websites
  - Degree, faculty size, student population, target market, …
  - Degree structure, individual course descriptions
  - Map between courses and SWEBOK
- Validated data with one or more professors from each school
- Analyzed for commonalities and uniqueness

Schools studied

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. DePaul University
8. Drexel University
9. *Dublin City University (Ireland) **
10. Embry-Riddle Aeronautical University
11. George Mason University
12. James Madison University
13. Mercer University
14. Monmouth University
15. Naval Postgraduate School
16. Penn State University – Great Valley
17. *Quebec University (Canada) **
18. Rochester Institute of Technology
19. Seattle University
20. Southern Methodist University
21. Stevens Institute of Technology
22. Texas Tech University
23. University of Alabama – Huntsville
24. University of Maryland University College
25. University of Michigan – Dearborn
26. University of Southern California
27. *University of York (UK) **
28. Villanova University

Non-US Schools

Observations from 28 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 *brittle*
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
6. Online course delivery is popular

More observations

7. Objective for graduates vary widely - software developer to researcher to software manager
8. Wide variation in depth and breadth of SWEBOK coverage in required and semi-required* courses
9. Many programs have required or semi-required courses that cover material that is either not in the SWEBOK at all or is not emphasized in the SWEBOK
10. Some significant topics are rarely mentioned - agility, software engineering economics, systems engineering
11. Some topics are ubiquitous - formal methods and architecture
12. "Object-oriented" is the standard development paradigm - creating a "clash" with many systems engineering programs that emphasize structured methods

*A student has a 50% or greater probability of taking a *semi-required* course.

School of Systems & Enterprises

Diverse focuses

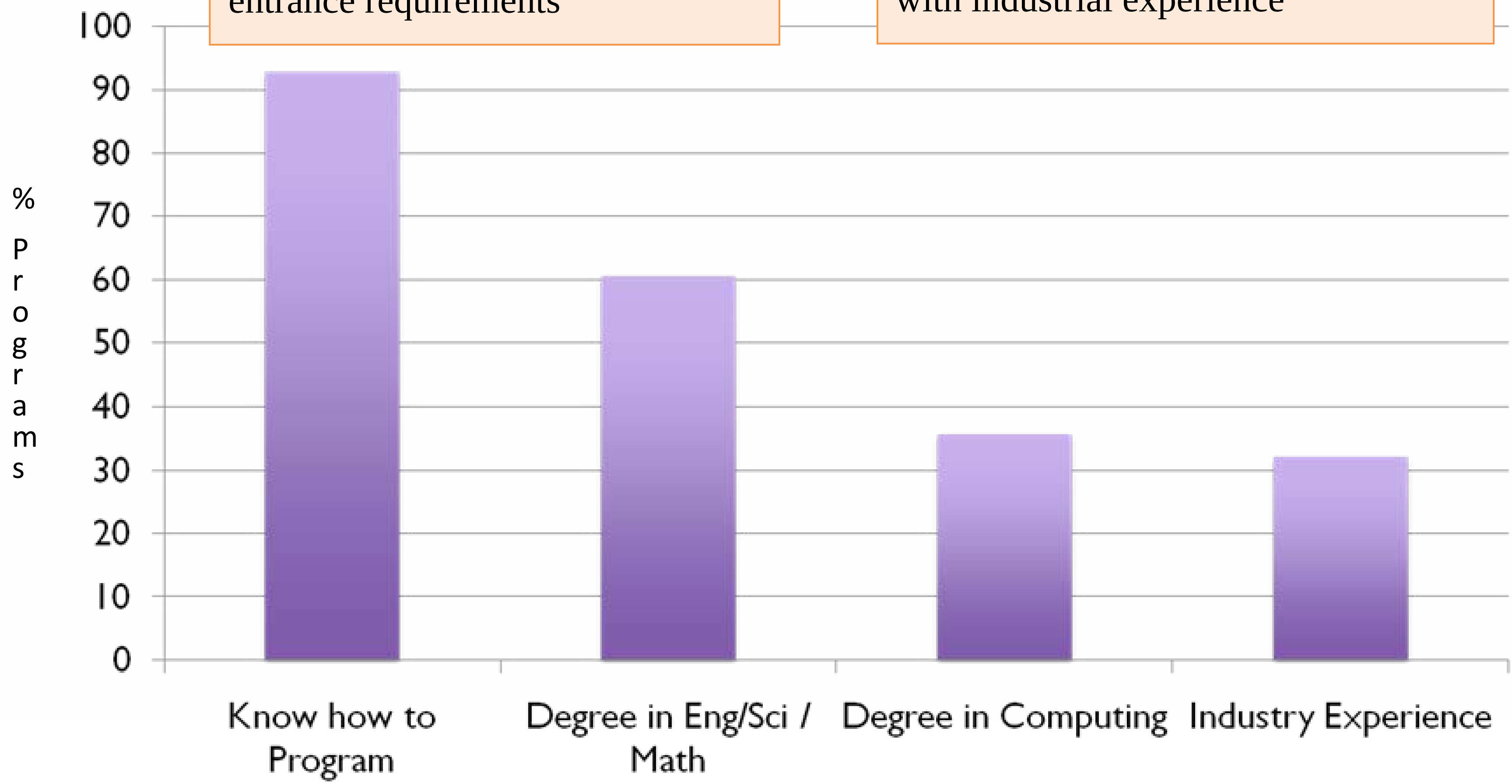
1. Development of defense systems
2. Acquisition of defense systems
3. Embedded real-time systems
4. Entrepreneurial technology companies
5. Quantitative software engineering
6. Software economics
7. Safety critical systems
8. Secure software engineering
9. Highly dependable software systems

No focus dominated

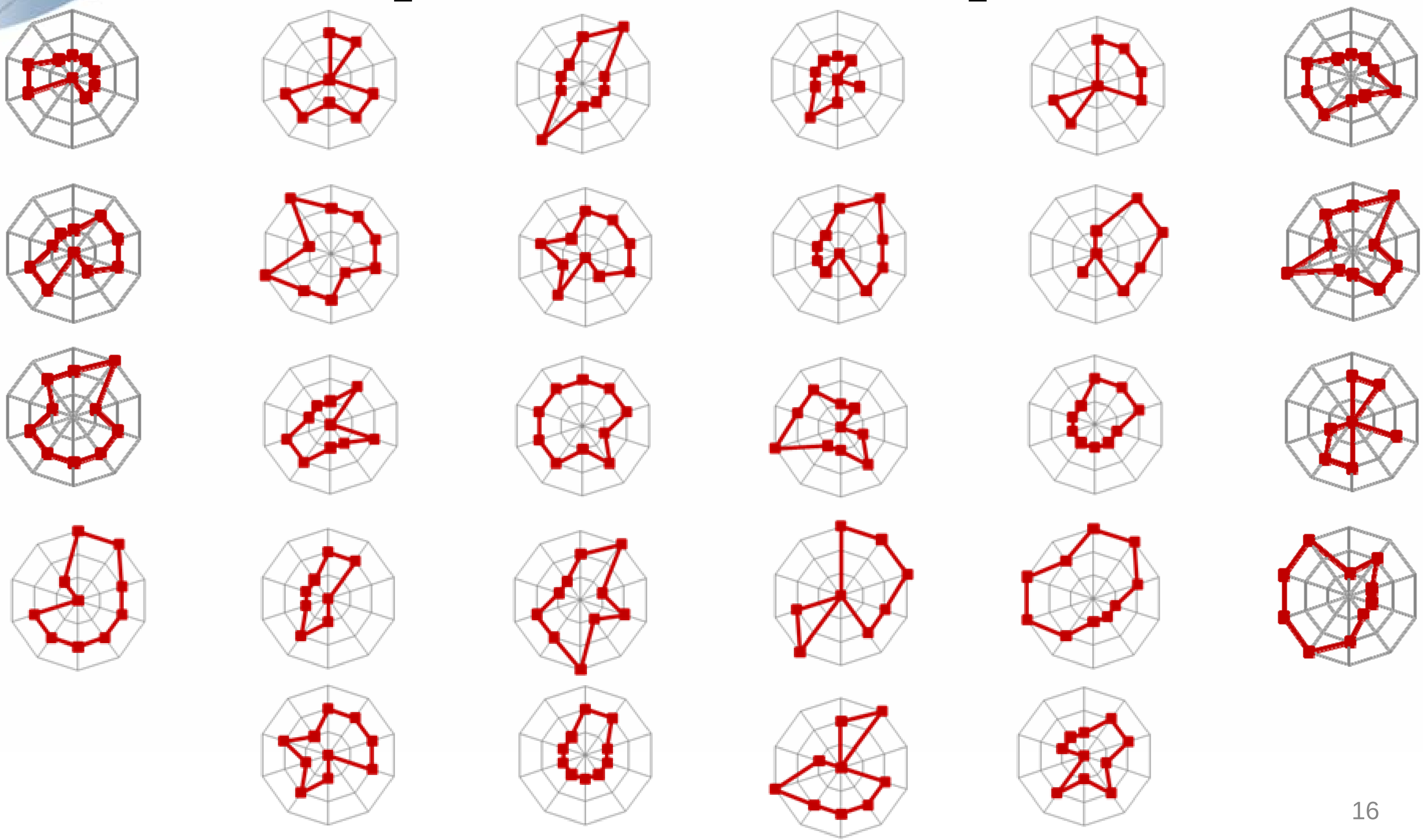
Entrance requirements

Most programs offer leveling courses for students lacking entrance requirements

Many programs routinely waive academic requirements for students with industrial experience



SWEBOK coverage in required and semi-required courses



The approach – GSwERC 0.50

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. Obtain endorsement from ACM, IEEE, INCOSE, NDIA, and other professional organizations (continuous)
5. ***Create GSwERC 0.50 suitable for broad community review and early adoption (October 2008)***
6. Create GSwERC 1.0 suitable for broad adoption (2009)

School of Systems & Enterprises

Expectations at entry

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

Outcomes 1 to 4 at graduation

1. Mastered the Core Body of Knowledge
2. Mastered at least one application domain, such as finance, medical, transportation, or telecommunications, and one application type, such as real-time, embedded, safety-critical, or highly distributed systems. 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. Mastered at least one knowledge area or sub-area from the CBOK to at least the Bloom Synthesis level.
4. Demonstrated how to make ethical professional decisions and practice ethical professional behavior.

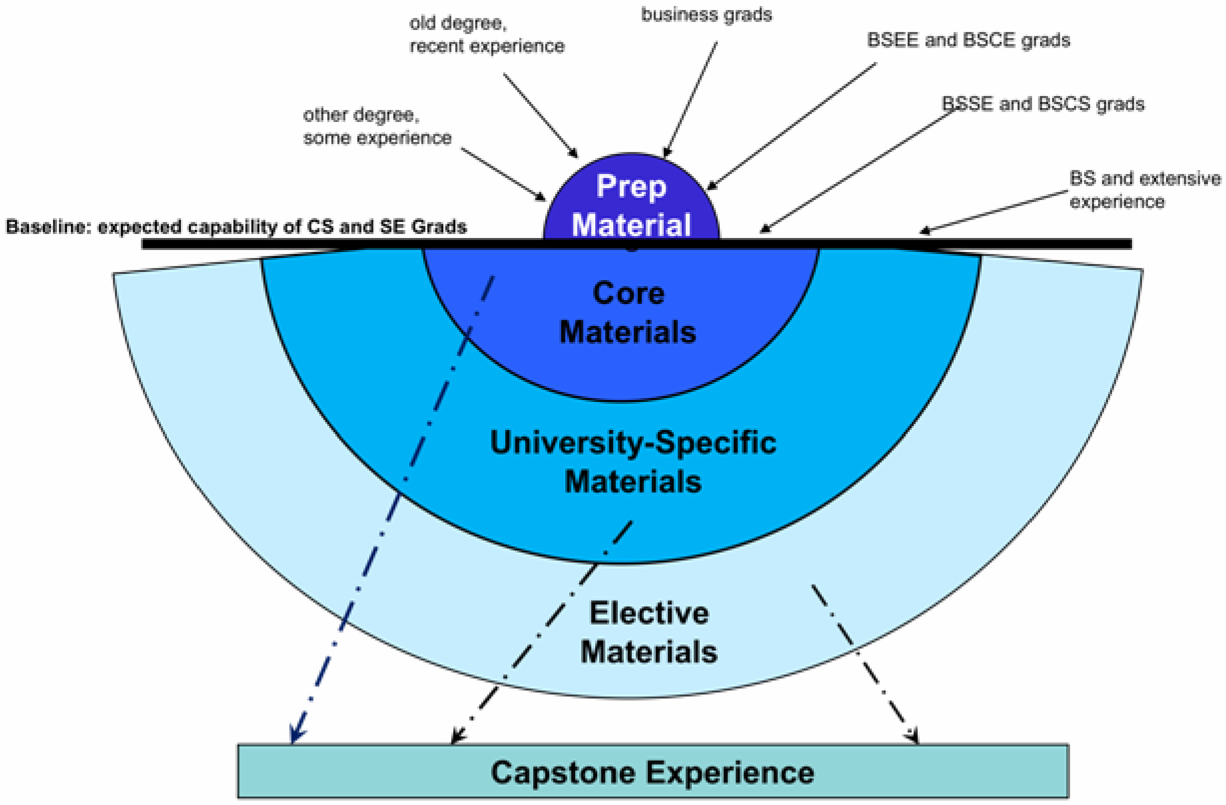
Outcomes 5 to 7 at graduation

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.
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, and to lead in one area of project development, such as project management, requirements analysis, architecture, construction, or quality assurance.
7. Show ability to reconcile conflicting project objectives, finding acceptable compromises within limitations of cost, time, knowledge, existing systems, and organizations.

Outcomes 8 to 10 at graduation

8. Understand and appreciate the importance of feasibility analysis, negotiation, effective work habits, leadership, and good communication with stakeholders in a typical software development environment.
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.

Curriculum architecture



Additional material in GSwERC

- Comparison of existing graduate software engineering programs with GSwERC recommendations – know how big the gap is between recommendations and practice
- Strategies recommended by the authors to implement GSwERC
- Hypothetical modifications of existing programs to more fully satisfy GSwERC

Reviewers, authors, and early adopters

SEEKING AUTHORS, REVIEWERS AND EARLY ADOPTERS FOR VERSION 0.5

GSwERC

A New Reference Curriculum for Graduate Studies Leading to a **Master's Degree in Software Engineering**

Since August 2007, a group of more than 30 professionals from academia, industry, and government have been developing a new reference curriculum leading to a Master's Degree in Software Engineering. The new reference curriculum will integrate systems engineering into the education of software engineers and reflect the dramatic changes in how software is used and developed since the last major graduate reference curriculum was published in the early 1990's. The effort is endorsed by the International Council on Systems Engineering (INCOSE), and the U.S. National Defense Industrial Association (NDIA) Systems Engineering Division. The IEEE Computer Society has a participating author and the ACM has a volunteer contributor. Sponsorship and funding for this effort are being provided by the U.S. Department of Defense.

Version 0.25 of the Graduate Software Engineering Reference Curriculum (GSwERC) was released in February 2008 for a limited review. Version 0.5 is being readied for world-wide release at the end of October 2008. The document will be posted on the GSwERC website (www.GSwERC.org). Review comments from all interested professionals are being sought. Version 1.0 baseline is expected sometime in 2009.

A clear limitation of the effort so far has been the demographics of the author team. While the team is extraordinarily talented and dedicated, the authors of these first two drafts were mainly from the United States. The first round of reviewers was more international, but still dominated by professionals from the U.S. In order to ensure that GSwERC has global applicability, we seek to further broaden both the reviewers for the second draft and the author team for Version 1.0.

One of the novel features of GSwERC is the inclusion of explicit comparisons of existing graduate software engineering programs to GSwERC recommendations and the inclusion of hypothetical modifications to two of those programs to better match GSwERC. These comparisons and modifications offer a window on how well GSwERC aligns with existing practice and will help faculty understand how to adopt GSwERC in their own universities. We welcome additional comparisons and hypothetical modifications from other universities to provide more insight into the gap between GSwERC and current practice and how to close that gap.

The Graduate Software Engineering Reference Curriculum (GSwERC) is ready for early adoption. Several of the current authors are now integrating portions of GSwERC in their own programs. We are also interested in helping universities begin to adopt GSwERC in 2009.

Please visit our website at
www.GSwERC.org
for more information.

Dr. Art Pyster, Distinguished Research Professor, from Stevens Institute of Technology in the United States, is the project leader for GSwERC.

Please contact him at **art.pyster@stevens.edu** if you are interested in participating.

STEVENS Institute of Technology

ASysT INSTITUTE

www.GSwERC.org